

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-94912

(P2002-94912A)

(43)公開日 平成14年3月29日 (2002.3.29)

(51)Int.Cl. ⁷	識別記号	F I	テ-マコト ⁸ (参考)
H 04 N 5/76		H 04 N 5/76	Z 5 B 0 8 9
G 06 F 11/00	3 3 0	G 06 F 11/00	3 3 0 A 5 C 0 1 8
13/00	3 5 7	13/00	3 5 7 A 5 C 0 5 2
17/60	1 7 6	17/60	1 7 6 A 5 C 0 5 3
H 04 N 5/765		H 04 N 5/782	K

審査請求 未請求 請求項の数10 O L (全 26 頁) 最終頁に続く

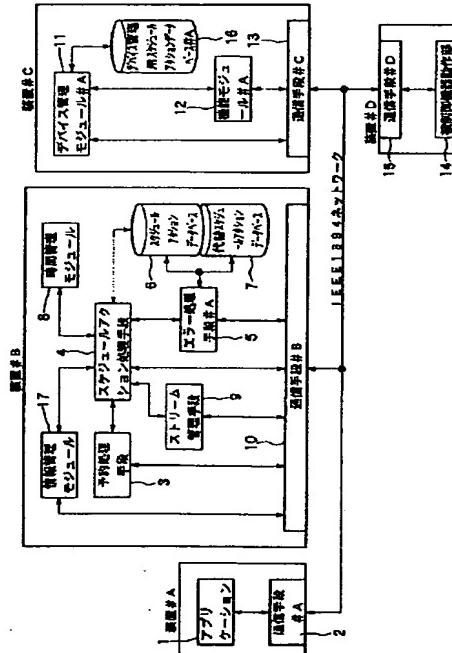
(21)出願番号	特願2000-278332(P2000-278332)	(71)出願人	000005049 シャープ株式会社 大阪府大阪市阿倍野区長池町22番22号
(22)出願日	平成12年9月13日(2000.9.13)	(72)発明者	小島 靖子 大阪府大阪市阿倍野区長池町22番22号 シ ャープ株式会社内
		(74)代理人	100091096 弁理士 平木 祐輔 Fターム(参考) 5B089 GA23 GB02 HA18 JA35 KA13 KC24 KC49 5C018 HA08 HA09 HA10 5C052 AA02 DD04 DD10 5C053 FA23 JA01 LA14

(54)【発明の名称】 リソース管理装置及びリソース管理方法

(57)【要約】 (修正有)

【課題】利用予約した機器に不具合があった場合、代替え案を示すことができるリソース管理装置及び方法を提供する。

【解決手段】アプリケーション実行装置#Aと、被制御機器内の機能を管理する機能モジュールの予約処理を行う予約処理手段3、コマンドを登録し実行開始するスケジュールアクション処理手段4、スケジュールアクションデータベース6を備える装置#Bと、被制御機器を管理するデバイス管理モジュール#A11と、被制御機器内の機能を管理する機能モジュール#A12を備える装置#Cが相互接続され、装置#Bはスケジュールアクション処理手段4によりエラーが検出された場合にエラー処理を行う手段#A5と、代替えスケジュールアクションデータベース7を備え、装置#Cはデバイス管理モジュール#A11に制限されたスケジュールアクションを保存しておくデバイス管理用スケジュールアクションデータベース#A16を備える。



【特許請求の範囲】

【請求項1】 ネットワーク上に存在する任意の装置のうち機能モジュールにより制御される第1の装置と、前記機能モジュールの予約処理を行う予約処理手段と、コマンドをあらかじめ登録しておき、コマンドの実行を開始するスケジュールアクション処理手段と、登録及び実行が行われるスケジュールアクションを保存しておくスケジュールアクションデータベースとを備える第2の装置と、

前記第1の装置を管理するデバイス管理手段と、前記第1の装置内の機能を管理する前記機能モジュールとを備える第3の装置と、

前記第1、第2及び第3の装置が通信回線によって相互に接続されて構成されるネットワークにおけるリソース管理装置において、

前記第2の装置は、

代替のスケジュールアクションを保存しておく代替スケジュールアクションデータベースと、

前記スケジュールアクション処理手段によってエラーが検出された場合に、前記代替スケジュールアクションデータベースを用いて代替スケジュールアクションを作成し、作成した代替スケジュールアクションと元のスケジュールアクションとを対応付けるエラー処理を行うエラー処理手段とを備え、

前記第3の装置は、

前記デバイス管理手段により制限されたスケジュールアクションを保存しておくデバイス管理用スケジュールアクションデータベースを備えることを特徴とするリソース管理装置。

【請求項2】 前記ネットワーク上に、モジュールを利用したアプリケーションを実行可能な第4の装置をさらに備えることを特徴とする請求項1記載のリソース管理装置。

【請求項3】 前記第2の装置は、

前記代替スケジュールアクションで指定された内容を記憶する一時記憶ディスクと、

前記機能モジュールの代替となる代替モジュールの検索と前記代替スケジュールアクションの作成及び登録と前記一時記憶ディスクにアクセスする機能及び予約と解放を行うエラー処理手段とを備えることを特徴とする請求項1記載のリソース管理装置。

【請求項4】 前記第2の装置は、

前記代替スケジュールアクションで指定された内容を記憶する一時記憶ディスク装置と、

前記一時記憶ディスク装置のデバイスを制御するデバイス管理モジュールと、

前記一時記憶ディスク装置の機能を制御する機能モジュールと、

前記デバイス管理モジュールのスケジュールアクションを記録しておくデバイス管理用スケジュールアクション

データベースと、

前記ネットワーク上に前記一時記憶ディスク装置が存在するかどうかを検索するエラー処理手段とを備えることを特徴とする請求項1記載のリソース管理装置。

【請求項5】 前記機能モジュールの予約処理を行う予約処理手段と、開始時間になると実行を開始するスケジュールアクション処理手段と、

スケジュールアクションを記録しておくスケジュールアクションデータベースと、

前記ネットワークに接続されている他の装置が持つエラー処理手段に対してエラー処理を依頼するエラー処理依頼手段とから構成される第5の装置をさらに備え、前記第2の装置は、前記ネットワークに接続されている任意の装置からのエラー処理依頼を受けてスケジュールアクションのエラー処理を行うエラー処理手段を備えることを特徴とする請求項1記載のリソース管理装置。

【請求項6】 前記機能モジュールの予約処理を行う予約処理手段と、

開始時間になると実行を開始するスケジュールアクション処理手段と、

スケジュールアクションを記録しておくスケジュールアクションデータベースと、前記ネットワークの接続されている他の装置の持つエラー処理手段に対してエラー処理を依頼するエラー処理依頼手段とから構成される第5の装置をさらに備え、前記一時記憶ディスク装置は、前記第2の装置の外部の前記ネットワークに接続されており、

前記第2の装置は、前記一時記憶ディスクを利用した代替スケジュールアクションを検索するエラー処理手段を備えることを特徴とする請求項3記載のリソース管理装置。

【請求項7】 複数のネットワーク装置が通信回線によって相互に接続されて構成されるネットワークにおけるリソース管理方法において、

利用者が予約したスケジュールアクションを登録するステップと、

登録されたスケジュールアクションを実行するステップと、

スケジュールアクションの実行状態を監視するステップと、

前記ステップで検出されたエラー情報をエラー処理手段に通知するステップと、

前記エラー処理手段では、検出されたエラーを受け取って代替スケジュールアクションデータベースに保存されている代替スケジュールアクションを検索するステップと、

検索した代替スケジュールアクションを登録するステップと、

登録した代替スケジュールアクションと元のスケジュー

3

ルアクションとを関連付けるステップとを順次実行することを特徴とするリソース管理方法。

【請求項8】 エラー処理手段の検索ステップでは、ネットワーク装置内部に備えられた一時記憶ディスクを利用して代替案を検索を行うことを特徴とする請求項7記載のリソース管理方法。

【請求項9】 エラー処理手段の検索ステップでは、複数のネットワーク装置の外部に備えられた一時記憶ディスクを利用して代替案の検索を行うことを特徴とする請求項7記載のリソース管理方法。

【請求項10】 エラー処理手段の検索ステップでは、他の装置からのエラー処理依頼を受けて代替案の検索を行うことを特徴とする請求項7記載のリソース管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、複数のネットワーク装置が通信回線によって相互に接続されて構成されるネットワークにおいて、被制御機器を制御する機能モジュールを複数の装置から利用するためのリソース管理装置及びリソース管理方法に関する。

【0002】

【従来の技術】 AV機器に代表されるようなネットワークを構成せずに利用される装置においては、自端末内に被制御手段とそれを制御するためのデバイス管理モジュール又は機能モジュールを持っている。図13は、ネットワークを構成しない制御機器の構成を示すブロック図である。図13において、100は制御機器、101はアプリケーション、102はデバイス管理モジュール、103は機能モジュール、104は被制御手段、105はスケジュールアクション処理を実行するときに用いられるスケジュールアクションデータベースである。

【0003】 ユーザはアプリケーション101で指定することによって、将来のある時間にこれらのモジュール102、103を利用することを予約することが可能である。VTR機器の録画予約や視聴予約などが例として挙げられる。装置によっては、他の装置をコントロールするコントローラを持ち、2台で動作を行うことが可能なものもある。しかし、装置によってあらかじめ決められた動作を行うことしかできず、あらかじめ用意された機器同士を接続することしかできない。また、テープが不足した場合など予約ができない場合は装置の動作を止めるだけである。そこで、複数の装置が通信回線によって相互に接続されたネットワークを構成する。このネットワーク上で共通のメッセージやコマンドを送受信することで、各装置内で持っているモジュールや機能を他の装置から利用することができる。

【0004】 従来、家庭内AVネットワークを実現するための技術として、例えば2000年1月に標準化が完了したHAVi (Home Audio/Video Interoperabilit

4

y) Architectureと呼ばれる標準仕様がある。この仕様は、HAVi V1.0 Specification版の概要部分 (1 General の1.1 Scope) に記述されているように、家庭用電化製品やコンピュータを接続して、ユーザがある機器を使って別の機器を操作するためのインターフェースの提供を実現している。HAVi仕様書では、一例としてIEEE (Institute of Electrical and Electronics Engineers) 1394とIEC (International Electrotechnical Commission) 61883準拠の家庭用電化製品によるネットワークの構築を想定している。また、同仕様書の同概要部分にあるように、HAViによって実現されたネットワークに接続されている機器をすべてのユーザが自由に使用できる。このように、家庭内にあるAV機器を接続してAVネットワークを構築することにより、ユーザは離れた部屋にある機器であっても自由に組み合わせて使用できるようになった。

【0005】 図14は、ネットワークによって構成される制御機器の構成を示すブロック図である。図14において、110はネットワークによって構成される第1の制御機器、111はアプリケーション、112はデバイス管理モジュール、113は機能モジュール、114は通信手段、115はスケジュールアクション処理を実行するときに用いられるスケジュールアクションデータベースである。また、120はネットワークによって構成される第2の制御機器、121はアプリケーション、122はデバイス管理モジュール、123は機能モジュール、124は通信手段、125はスケジュールアクション処理を実行するときに用いられるスケジュールアクションデータベースである。また、130は第1の被制御機器、131は通信手段、132は被制御機器動作部であり、140は第1の被制御機器、141は通信手段、142は被制御機器動作部である。

【0006】 上記各機器は、IEEE1394規格に準拠した通信回線によって相互に接続されてネットワークを構成する。HAViには、リソース管理用のモジュールが定義されており、開始時間・停止時間だけでなく、利用するモジュールや送信するコマンドを指定することによって、あらかじめ決められた動作や機器の接続だけに限らず、さまざまなコマンドを動作させたり、いろいろな機器のストリームを接続させることができる。アプリケーションを起動させることも可能である。HAViでは、開始時間・停止時間・スケジュールアクションの実行に必要なモジュール・実行されるべき一連のコマンド・接続リスト・実行にアプリケーションなどが必要な場合は、そのIDなどを示したスケジュールアクションを登録し、開始時間に自動的に実行させることができる。登録の際には、必要なモジュールがあるかどうか、実行が可能かどうか、接続が有効かどうかなど実現可能性のチェックを行い、実現可能な場合に登録が行われる。さらに、登録されてから開始時間になるまで、必要

なモジュールがあるかどうか、ネットワークリソースが十分か否かについて常にチェックを行っている。途中でデバイスがなくなった場合や、ネットワークリソースが足りなくなった場合には、スケジュールアクションを無効にし、無効になったことを示すイベントを発行する。

【0007】さらに、特開平10-308776号公報には、リソース管理サーバを設け、簡単な操作でネットワークリソースを予約できるようにしたネットワークリソースの予約方式が開示されている。公報に記載のリソースの予約方式では、リソース予約に失敗した場合は代替案を自動的に見つけ出して提示する。

【0008】

【発明が解決しようとする課題】しかしながら、このような従来のリソース管理装置にあっては、スケジュールアクションの登録から実行終了までの間に、ネットワークの状態が変化して実行条件を満たさなくなりエラーが検出された場合に、エラーを回避するための手段を持たないという問題点があった。例えば、HAVIにおいてはスケジュールアクションが実行できなくなった場合イベントを発行するだけであり、検出されたエラーについてエラーを回避するためのリカバリ手段を持たない。したがって、スケジュールアクションを登録したユーザにとって、本当にスケジュールアクションの実行が行われるのかどうかが保証されない。また、上記公報に記載の装置は、ネットワークリソースの予約時に予約が拒絶された場合に代替案を提案するものであって、すでに登録された予約については何かの都合で実行されなくなても代替案が提示されないという課題があった。

【0009】また、代替案を提示してユーザが選択すると、元の予約は記憶されず、代替案しか登録を行うことはできない。そのため、元の予約が実現可能になった場合でも、代替案から元の予約に戻す手段がないという課題があった。本発明は、このような課題に鑑みてなされたものであって、利用者が予約した機器に不具合があった場合に、代替案を示すことができるリソース管理装置及びリソース管理方法を提供することを目的としている。

【0010】

【課題を解決するための手段】本発明のリソース管理装置は、ネットワーク上に存在する任意の装置のうち機能モジュールにより制御される第1の装置と、前記機能モジュールの予約処理を行う予約処理手段と、コマンドをあらかじめ登録しておき、コマンドの実行を開始するスケジュールアクション処理手段と、登録及び実行が行われるスケジュールアクションを保存しておくスケジュールアクションデータベースとを備える第2の装置と、前記第1の装置を管理するデバイス管理手段と、前記第1の装置内の機能を管理する前記機能モジュールとを備える第3の装置と、前記第1、第2及び第3の装置が通信回線によって相互に接続されて構成されるネットワーク

におけるリソース管理装置において、前記第2の装置は、代替のスケジュールアクションを保存しておく代替スケジュールアクションデータベースと、前記スケジュールアクション処理手段によってエラーが検出された場合に、前記代替スケジュールアクションデータベースを用いて代替スケジュールアクションを作成し、作成した代替スケジュールアクションと元のスケジュールアクションとを対応付けるエラー処理を行うエラー処理手段とを備え、前記第3の装置は、前記デバイス管理手段により制限されたスケジュールアクションを保存しておくデバイス管理用スケジュールアクションデータベースを備えることを特徴としている。

【0011】また、好ましくは、前記ネットワーク上に、モジュールを利用したアプリケーションを実行可能な第4の装置をさらに備えるものであってもよい。また、前記第2の装置は、前記代替スケジュールアクションで指定された内容を記憶する一時記憶ディスクと、前記機能モジュールの代替となる代替モジュールの検索と前記代替スケジュールアクションの作成及び登録と前記20一時記憶ディスクにアクセスする機能及び予約と解放を行うエラー処理手段とを備えることで、利用者が予約した機器に不具合があった場合に、一時記憶ディスクを含む代替案を示すことができる。

【0012】また、前記第2の装置は、前記代替スケジュールアクションで指定された内容を記憶する一時記憶ディスク装置と、前記一時記憶ディスク装置のデバイスを制御するデバイス管理モジュールと、前記一時記憶ディスク装置の機能を制御する機能モジュールと、前記デバイス管理モジュールのスケジュールアクションを記録30しておくデバイス管理用スケジュールアクションデータベースと、前記ネットワーク上に前記一時記憶ディスク装置が存在するかどうかを検索するエラー処理手段とを備えることで、エラー処理手段を持つ端末内に十分な記憶容量が確保できなくても代替スケジュールアクションが処理可能であり、端末のデータ記憶領域が圧迫されない。また、いくつかのスケジュールアクションを同時に処理することが可能となる。

【0013】また、前記機能モジュールの予約処理を行う予約処理手段と、開始時間になると実行を開始するスケジュールアクション処理手段と、スケジュールアクションを記録しておくスケジュールアクションデータベースと、前記ネットワークに接続されている他の装置が持つエラー処理手段に対してエラー処理を依頼するエラー処理依頼手段とから構成される第5の装置をさらに備え、前記第2の装置は、前記ネットワークに接続されている任意の装置からのエラー処理依頼を受けてスケジュールアクションのエラー処理を行うエラー処理手段を備えることで、ネットワーク上にあるすべてのスケジュールアクション処理手段がエラー処理手段及び代替スケジュールアクションデータベースを持つ必要がなく、プロ

グラムの効率が良い。

【0014】また、前記機能モジュールの予約処理を行う予約処理手段と、開始時間になると実行を開始するスケジュールアクション処理手段と、スケジュールアクションを記録しておくスケジュールアクションデータベースと、前記ネットワークの接続されている他の装置の持つエラー処理手段に対してエラー処理を依頼するエラー処理依頼手段とから構成される第5の装置をさらに備え、前記一時記憶ディスク装置は、前記第2の装置の外部の前記ネットワークに接続されており、前記第2の装置は、前記一時記憶ディスクを利用した代替スケジュールアクションを検索するエラー処理手段を備えることで、利用者が予約した機器に不具合があった場合でも多様な代替機器を示すことができる。

【0015】本発明のリソース管理方法は、複数のネットワーク装置が通信回線によって相互に接続されて構成されるネットワークにおけるリソース管理方法において、利用者が予約したスケジュールアクションを登録するステップと、登録されたスケジュールアクションを実行するステップと、スケジュールアクションの実行状態を監視するステップと、前記ステップで検出されたエラー情報をエラー処理手段に通知するステップと、前記エラー処理手段では、検出されたエラーを受け取って代替スケジュールアクションデータベースに保存されている代替スケジュールアクションを検索するステップと、検索した代替スケジュールアクションを登録するステップと、登録した代替スケジュールアクションと元のスケジュールアクションとを関連付けるステップとを順次実行することを特徴としている。

【0016】また、エラー処理手段の検索ステップでは、ネットワーク装置内部に備えられた一時記憶ディスクを利用して代替案を検索を行うものであってもよい。また、エラー処理手段の検索ステップでは、複数のネットワーク装置の外部に備えられた一時記憶ディスクを利用して代替案の検索を行うものであってもよい。また、エラー処理手段の検索ステップでは、他の装置からのエラー処理依頼を受けて代替案の検索を行うものであってもよい。

【0017】

【発明の実施の形態】以下、添付図面を参照しながら本発明の好適なリソース管理装置の実施の形態について詳細に説明する。

【0018】第1の実施の形態

図1は、本発明の第1の実施の形態のネットワークシステムの全体構成を示す図である。図1において、IEEE 1394に準拠した通信回線によって相互に接続されたネットワーク上に、装置#A（第4の装置）、装置#B（第2の装置）、装置#C（第3の装置）、装置#D（第1の装置）がある。

【0019】装置#Aは、ネットワーク上に存在する裝

置のうち必要なモジュールを利用したアプリケーションを実行できるアプリケーション1、装置#A内でアプリケーション1から送信されたメッセージ又はコマンドをIEEE 1394ネットワーク上に配信するためのデータ変換及び、ネットワーク上から受け取ったメッセージやコマンドを装置#A内のメッセージ又はコマンドに変換するための通信手段#A2を備える。通信手段#A2はまた、ネットワークの変化を検出したり、データの送受信の管理を行う。以下、ネットワークと示した場合には、IEEE 1394ネットワークのことと指す。

【0020】装置#Bは、利用するモジュールのために機能モジュールの予約処理を行う予約処理手段3、開始時間・停止時間・必要なモジュール・送信するコマンドをあらかじめ登録しておき、実行可能な状態を監視しておき、開始時間になると実行を開始するスケジュールアクション処理手段4、スケジュールアクション処理手段4によって登録や実行が行われるスケジュールアクションを保存しておくスケジュールアクションデータベース6、開始時間になる前にスケジュールアクションの実行に必要なモジュールがなくなった場合や実行に失敗した場合など実行条件を満たさないことにより、スケジュールアクション処理手段4によってエラーが検出された場合に、代替スケジュールアクションを作成し元のスケジュールアクションと対応付けを行なうなどエラー処理を行うためのエラー処理手段#A5、代替スケジュールアクションを保存しておく代替スケジュールアクションデータベース7、タイムを管理する時間管理モジュール8、モジュール間及び装置間のストリームの接続を行うストリーム管理手段9、ネットワーク上のモジュール情報を管理している情報管理モジュール17、装置#B内のデータについてデータ変換を行いデータの送受信を管理する通信手段#B10を備える。

【0021】装置#Cは、被制御機器を管理するデバイス管理モジュール#A11（デバイス管理手段）、被制御機器内の機能を管理する機能モジュール#A12、デバイス管理モジュール#A11に制限されたスケジュールアクションを保存しておくデバイス管理用スケジュールアクションデータベース#A16、装置#C内のデータについてデータ変換を行いデータの送受信を管理する通信手段#C13を備える。

【0022】装置#Dは、デバイス管理モジュール#A11、機能モジュール#A12により制御される被制御機器動作部14、装置#D内のデータについてデータ変換を行い、データの送受信を管理する通信手段#D15を備える。通信手段#B10、通信手段#C13、通信手段#D15は、通信手段#A2と同様に、自端末内のメッセージ又はコマンドをネットワーク上に配信するためのデータ変換を行い、ネットワーク上から受け取ったメッセージやコマンドを自端末内のメッセージ又はコマンドに変換するための機能を持つ。また、ネットワーク

に関するイベントや変化を検出する機能を持つ。装置#A、装置#B、装置#C間のメッセージのやりとりは、あらかじめ定義されたデータ形式で行われるものとする。

【0023】本実施の形態ではメッセージを送信するということは、定義されたデータ形式のメッセージを送信することを表すものとする。また、装置#C、装置#D間のメッセージ又はコマンドのやりとりは、デバイス管理モジュール#A11及び機能モジュール#A12と、被制御機器動作部14の間で相互に理解可能な形式で行われるものとする。被制御機器に関わる命令を示したものとコマンドと表す。それぞれの装置内でのコマンドのやりとりは端末固有であり、データ形式は問わない。

【0024】スケジュールアクションとは、開始時間・停止時間・スケジュールアクションの実行に必要なモジュール・実行されるべき一連のコマンド・接続リスト・実行にアプリケーションなどが必要な場合はそのIDなどが示されたものである。開始時間になると、必要なモジュールの予約を行い、接続リストに従ってストリーム接続を行い、一連のコマンドを実行する。登録したアプリケーションがネットワーク上に存在しなくても決められた動作を行うことができる仕組みになっている。停止時間になると、一連のコマンドを実行し、ストリームを切断し、予約されたモジュールを解放する。停止時間をすぎると、データは削除される。

【0025】まず、アプリケーション1は、ユーザとのインターフェース手段を持ち、ユーザからの指定によってスケジュールアクション1のデータを作成する。スケジュールアクション処理手段4に対してスケジュールアクション1を登録する。アプリケーション1から送信されたメッセージは、通信手段#A2でネットワーク上のデータに変換され、通信手段#B10に配信される。通信手段#B10は装置#B内のデータに変換し、スケジュールアクション処理手段4にメッセージを送信する。装置間でやりとりされるメッセージについてはすべて通信手段でデータ変換及びデータ送受信の処理が行われるが、以降の説明では省略する。

【0026】スケジュールアクションのデータには、開始時間・停止時間・スケジュールアクションの実行に必要なモジュール・実行されるべき一連のコマンド・接続リスト・実行にアプリケーションなどが必要な場合はそのIDなどが含まれる。スケジュールアクション処理手段4は、依頼のあったスケジュールアクション1について、必要なモジュールが該当時間に利用可能かどうか、コマンドはすべて実行可能か、接続リストはすべて実行可能かを関係するデバイス管理モジュールに問い合わせる。関係するデバイス管理モジュールは、必要なモジュールリストに示されている。

【0027】スケジュールアクション1のモジュールリストには、デバイス管理モジュール#A11が含まれて

いるものとする。デバイス管理モジュール#A11には、自身の関係する情報のみが渡される。登録が可能であれば、デバイス管理モジュール#A11は、自身のデバイス管理用スケジュールアクションデータベース#A16に情報を保持する。デバイス管理モジュール#A11から登録可能との返事が戻ってくると、スケジュールアクション処理手段4は該当時間に重なっているスケジュールアクションのリストを他のスケジュールアクション処理手段に問い合わせる。すべてのリストを検索し終わると、ネットワーク容量の情報を取得し、スケジュールアクションの実行に必要なストリーム接続を行うためのネットワーク容量が十分かどうか確認し、容量を越えなければスケジュールアクションデータベース6に登録を行う。さらに、アプリケーション1に登録に成功したことを返信する。満たさなければ登録を中止し、すべての登録を削除してから失敗を返信する。

【0028】以下、上述のように構成されたネットワークシステムにおけるリソース管理装置の動作を説明する。図2及び図3は、スケジュールアクションの登録の20実行の動作を示すフローチャートである。図中、Sはフローのステップを示す。アプリケーション1は、スケジュールアクション1を作成し、スケジュールアクション処理手段4に対してスケジュールアクション1の登録を行う(ステップS1)。登録は前述した方法で行われる。スケジュールアクション1は、デバイス管理モジュール#A11、機能モジュール#A12を含むものとする。スケジュールアクション処理手段4は、登録が完了したスケジュールアクション1について被制御機器、デバイス管理モジュール、機能モジュールを監視する(ステップS2)。監視とは、ネットワーク上に存在しているかどうかを常に見ておくことであり、ネットワーク上に存在しなくなると、その情報を検出する。

【0029】ここで、スケジュールアクション処理手段4は、機能モジュールのIDと種類を対応付けて覚えておく(ステップS3)。機能モジュールのIDは、スケジュールアクションデータに含まれている。種類は情報管理モジュール17から検索する。情報管理モジュール17は他の装置に存在していてもよい。また、並行してネットワーキングリソースの監視を行う(ステップS4)。ネットワーキングリソースとはネットワーク容量、チャネル数及び番号のことを表す。次いで、ステップS5で開始時間のタイマを設定する。装置#B内の時間管理モジュール8にタイマ設定を依頼するメッセージを送信することで設定する。又はネットワーク上にタイマ機能を持つモジュールがあれば、それを利用することもできる。

【0030】スケジュールアクション1の登録から開始時間までは、スケジュールアクション処理手段4はエラーの検出を行う(ステップS6)。上記ステップS2、50ステップS4においてモジュールの監視及びネットワー

11

クリソースの監視を行っているので、スケジュールアクション1が実行できないような状態になれば検出を行うことが可能である。エラーが検出されず、開始トリガーを受信するまで（ステップS7）は、開始トリガーが来るまで監視を続ける（ステップS8）。もし、エラーが検出されれば（ステップS6）、監視時エラー処理（ステップS9）を行う。監視時エラー処理については後述する。開始トリガーとは、タイマから時間が来たことを知らせる通知メッセージを受け取ることである。タイマ設定でなく、ある条件を満たしたら開始するというような指定がされていた場合には、条件を満たしたことを探すメッセージを受け取る。時間管理モジュール8にタイマを設定しておくと、そこからメッセージが送信される。開始トリガーを受け取ったら、スケジュールアクション処理手段4はスケジュールアクションの実行を開始する。

【0031】まず、必要なモジュールの予約を行う（ステップS10）。実行されるべき一連のコマンドには、対象モジュールのID、呼ばれるAPIの識別子を表すオペレーションコード、引数を表すデータが含まれる。対象モジュールのIDは分かるので、そこからデータの送り先のモジュールの予約を行う。機能モジュールの予約を行うため、スケジュールアクション処理手段4は、予約するべきモジュールのIDとともに、予約処理手段3に予約を依頼する。予約処理手段3は、対象モジュールに予約メッセージを送信し、結果を受け取る。すべての対象モジュールが予約可能な場合のみ予約を行うことができ、予約可能でなければ、すべての対象モジュールの予約を解放して、スケジュールアクション処理手段に失敗を返す。機能モジュールの予約とは、機能モジュールの状態変化を伴うようなコマンドの使用を予約対象者のみに制限するものである。例えば、アプリケーション1が機能モジュール#A12を予約したとする。このとき、アプリケーション1だけが機能モジュール#A12の状態を変更するコマンドを実行させることができる。その他のアプリケーションはコマンドを送信してもエラーが返信され、コマンドは実行されない。このようにすることで、機能モジュールの利用者が衝突しないようになる。

【0032】スケジュールアクション処理手段4は、機能モジュールの予約が成功すると（ステップS11）、接続リストからストリームの接続を行う（ステップS12）。スケジュールアクション処理手段4がストリーム管理手段9に対して、接続リストに従ってストリーム接続を依頼するメッセージを送信する。ストリーム管理手段9は、接続元モジュールと接続先モジュールのチャンネルを選択し、指定されたフォーマットのデータを指定して接続処理を行う。

【0033】接続に成功すると（ステップS13）、開始コマンドの送信を行う（ステップS14）。開始コマ

12

ンドとは開始時間に実行されるべき一連のコマンドのことであり、送信先のモジュール情報と対になって、スケジュールアクションのデータに含まれている。スケジュールアクション処理手段4は、開始コマンドとして指定されたデータを指定されたモジュールに送信する。開始コマンドはデータが並んでいる順番に送信される。一つのコマンドを送信し成功の返事が戻ってくるまで、次のコマンドは送信されない。開始コマンドの送信に成功すると（ステップS15）、スケジュールアクションの開始に成功する。機能モジュールの予約に失敗した場合（ステップS11）、又はストリームの接続に失敗した場合（ステップS13）、又はコマンドの送信に失敗した場合（ステップS15）、開始時エラー処理を行う（ステップS17）。開始時エラー処理の詳細については、図5により後述する。

【0034】スケジュールアクションの開始に成功すると、停止時間のタイマを設定する（ステップS16）。開始時間のタイマ設定と同様に、時間管理モジュール8にタイマ設定を依頼するメッセージを送信することで設定を行うことができる。スケジュールアクションを開始し停止するまでの間、スケジュールアクション処理手段4はエラーの検出を行う（ステップS18）。停止トリガーが来るまでエラーの監視を続ける（ステップS20）。もし、エラーが検出されれば（ステップS18）、実行時エラー処理を行う（ステップS21）。実行時エラー処理については後述する。

【0035】停止トリガーを受け取ったら（ステップS19）、停止コマンドを送信する（ステップS22）。停止コマンドとは開始コマンドと同様に停止時間に実行されるべき一連のコマンドのことであり、送信先のモジュール情報とともにスケジュールアクションのデータに含まれている。スケジュールアクション処理手段4は、停止コマンドとして指定されたデータを指定されたモジュールに送信する。一つのコマンドを送信し成功の返事が戻ってくるまで、次のコマンドは送信されない。しかし、停止コマンドについては、コマンドの実行に失敗しても次のコマンドは送信され、停止処理は止めないようにする。

【0036】すべての停止コマンドの送信が終わると、接続されているストリームの切断処理を行う（ステップS23）。スケジュールアクション処理手段4がストリーム管理手段9に対して、接続中のストリームの切断を依頼するメッセージを送信する。ストリーム管理手段9は、指定されたストリームの切断処理を行い結果を返す。ここで切断に失敗しても処理は続けられる。

【0037】すべてのストリームの切断処理が終わると、予約していたモジュールの解放を行う（ステップS24）。スケジュールアクション処理手段4は、予約処理手段3に対し、予約解放を依頼するメッセージを送信する。予約処理手段3は、指定された対象モジュールに

13

予約解放メッセージを送信し、結果を受け取る。すべてのモジュールから返事が戻ってくるとスケジュールアクション処理手段4に結果を返す。

【0038】以上の処理によってスケジュールアクションの停止が完了するので、デバイス管理モジュール#A11にスケジュールアクション1の削除を依頼し、スケジュールアクション1をスケジュールアクションデータベース6から削除して処理を終了する（ステップS25）。デバイス管理モジュール#A11は自身のデバイス管理用スケジュールアクションデータベース#A16に登録されたスケジュールアクション1を削除する。以上の処理を実行することにより、スケジュールアクションの監視時にエラーを検出した場合に、エラーを回避するための代替スケジュールアクションを作成することが可能となる。

【0039】次に、監視時エラー処理の詳細について述べる。図4は、監視時エラー処理の動作を示すフローチャートである。図2のステップS6においてエラーが検出された場合に本監視時エラー処理が行われる（ステップS9）。まず、エラー処理手段#A5は、スケジュールアクション処理手段4において検出されたエラーについてデバイス制御モジュール又は機能モジュールがなくなったことを示すかどうか調べる（ステップS26）。デバイス管理モジュールがなくなった場合にはそのデバイス上の機器を動作させるための機能モジュールもなくなることになる。上記モジュールがなくなった場合、エラーが検出されたスケジュールアクションの開始コマンドのデータから内容を解析する（ステップS27）。

【0040】スケジュールアクション処理手段4は、ステップS3において機能モジュールのIDと種類を対応付けしているので、エラー処理手段#A5は、スケジュールアクション処理手段4の情報を利用し対象モジュールのID、呼ばれるAPI(Application Program Interface)の識別子を表すオペレーションコードから、引数を表すデータを解析する。解析されたコマンドから、スケジュールアクションを実行させるために必要な機能モジュールを抽出する。例えば、テレビのチャンネルを設定してVTRに録画するようなスケジュールアクションを想定する。VTRを制御するためのVTR用機能モジュールがなくなった場合、テレビ番組の録画が可能な機能モジュールが抽出される。

【0041】次いで、ステップS28で消失した機能モジュールの代替となるモジュールを検索する。情報管理モジュール17に対して、検索対象のモジュールの機能を指定して検索依頼のメッセージを送信することで検索可能である。検索に成功した場合（ステップS29）、元のスケジュールアクション1についてなくなった機能モジュールを代替となる機能モジュールに置換えた代替スケジュールアクション2を作成する（ステップS30）。元のスケジュールアクション1を無効にし、デバ

14

イス管理モジュール#A11にスケジュールアクション1の削除を依頼する（ステップS31）。デバイス管理モジュール#A11は、自身のデバイス管理用スケジュールアクションデータベース#A16に登録されたスケジュールアクション1を削除する。

【0042】次いで、ステップS32でエラー処理手段#A5は代替スケジュールアクション2を登録する。登録方法はスケジュールアクション1と同様である。登録が成功すると（ステップS33）、元のスケジュールアクション1との対応付けを行う（ステップS34）。代替スケジュールアクション2のデータに代替かどうかを示すフラグと元のスケジュールアクションのIDを指すデータ領域を追加する。元のスケジュールアクション1には、代替スケジュールアクションがあることを示すフラグと、代替スケジュールアクション2のIDを指すデータを追加する。元のスケジュールアクション1が無効かどうかのフラグを立て無効であることを示す。

【0043】スケジュールアクション処理手段4は、元のスケジュールアクション1の監視は無効フラグが立っている状態でも同じように行われる。代替スケジュールアクション2について新たに被制御機器、デバイス管理モジュール、機能モジュールの監視、ネットワークリソースの監視を行う（ステップS35）。エラー処理手段#A5は、検出されたエラーがデバイス制御モジュール又は機能モジュールがなくなったことを示すものではない場合（ステップS26）、ネットワークリソースが原因でエラーになったことが考えられるため、ネットワーク容量が減少したかどうかを調べる（ステップS36）。ネットワーク容量が減少した場合、同じ時間内にスケジュールされているスケジュールアクションを検索する（ステップS37）。スケジュールアクション処理手段4の自身のスケジュールアクションデータベース6内にあるもの、及びネットワーク上の他のスケジュールアクション処理手段に依頼を送ることで検索を行う。また、デバイスがなくなった場合と同様に開始コマンドのデータから内容を解析する（ステップS38）。スケジュールアクションに含まれる機能モジュールのそれぞれについて、代替となるモジュールが自端末内にあるかどうか検索を行う（ステップS39）。以降の処理はデバイスがなくなった場合の処理と同様である。

【0044】ネットワーク容量が減少したのでない場合（ステップS36）、エラー処理手段#A5ではエラー処理を行うことができないため、代替処理不可と判定する（ステップS40）。スケジュールアクション1を無効にし、デバイス管理モジュール#A11への登録を削除し、無効フラグを立てる（ステップS41）。デバイス管理モジュール#A11は、自身のデバイス管理用スケジュールアクションデータベース#A16に登録されたスケジュールアクション1を削除する。無効になったスケジュールアクション1については、スケジュールア

15

クション処理手段4において引き続き、監視を続ける。【0045】代替となる機能モジュールの検索に失敗した場合(ステップS29)、及び代替スケジュールアクション2の登録において登録に失敗した場合(ステップS33)、エラー処理手段#A5ではエラー処理を行うことができないため、代替処理不可と判定する(ステップS42)。スケジュールアクション1を無効にし、デバイス管理モジュールへの登録を削除し、無効フラグを立てる(ステップS43)。デバイス管理モジュール#A11は、自身のデバイス管理用スケジュールアクションデータベース#A16に登録されたスケジュールアクション1を削除する。無効になったスケジュールアクション1については、スケジュールアクション処理手段4において引き続き監視を続ける。以上の監視時エラー処理を実行することにより、デバイス制御モジュールがなくなった場合に消失した機能モジュールの代替となるモジュールを検索して代替スケジュールアクションを作成することが可能になる。

【0046】次に、開始時エラー処理の詳細について述べる。図5は、開始時エラー処理の動作を示すフローチャートである。図2のステップS11、ステップS13、ステップS15においてエラーが検出された場合に、エラー処理手段#A5において本開始時エラー処理が行われる。

【0047】まず、どの段階でエラーが検出されたか調査する。モジュールの予約に失敗した場合(ステップS44)、予約に失敗したモジュールを利用させてもらえないかどうか、現在利用している利用者に対して、交渉を行うためのネゴシエーションメッセージを送信する(ステップS45)。ネゴシエーションの交渉に成功した場合(ステップS46)、モジュールの予約を行い、ステップS12に戻って処理を続ける。失敗した場合(ステップS46)、予約に失敗した機能モジュールの代替となるモジュールを検索する(ステップS47)。代替モジュールの検索方法は、監視時エラー処理の際の検索方法と同様に行うことが可能である。検索に成功した場合(ステップS48)、ステップS49～ステップS53までの処理は、ステップS30～ステップS34までの処理と同じである。

【0048】代替スケジュールアクション2について、すでに実行時間は過ぎているので、スケジュールアクションの実行処理を行う(ステップS54)。実行処理とは、ステップS10からの処理を行うことである。登録に失敗した場合(ステップS52)、エラー処理手段#A5ではエラー処理を行うことができないため、代替処理不可と判定する(ステップS61)。スケジュールアクションの停止処理を行う(ステップS62)。スケジュールアクション1は開始時間をすぎているので、デバイス管理モジュールにスケジュールアクション1の削除を依頼し、削除したことを見知らせるためのイベントを発

16

行し、関係するデバイス管理モジュールにスケジュールアクションの削除を依頼し、スケジュールアクション1をスケジュールアクションデータベース6から削除する(ステップS63)。

【0049】上記ステップS44でモジュールの予約に失敗したのではなく、接続に失敗した場合(ステップS55)、失敗した接続条件でもう一度メッセージを送信し、再接続を試みる(ステップS56)。再接続に成功した場合には(ステップS57)、S14からの処理を続ける。失敗した場合(ステップS57)、代替処理不可と判定し(ステップS61)、スケジュールアクションの停止処理を行う(ステップS62)。スケジュールアクション1をスケジュールアクションデータベース6から削除する(ステップS63)。

【0050】上記ステップS58でコマンドの送信に失敗した場合、コマンドの再送信を試みる(ステップS59)。再送信に成功した場合には(ステップS60)、S16からの処理を続ける。失敗した場合(ステップS60)、代替処理不可と判定し(ステップS61)、スケジュールアクションの停止処理を行う(ステップS62)。関係するデバイス管理モジュールにスケジュールアクション1の削除を依頼し、スケジュールアクション1をスケジュールアクションデータベース6から削除する(ステップS63)。以上の開始時エラー処理を実行することにより、スケジュールアクションの開始時にエラーを検出した場合、実行に失敗したスケジュールアクションの代わりとなる代替スケジュールアクションを作成し実行することが可能になる。

【0051】次に、実行時エラー処理の詳細について述べる。図6は、実行時エラー処理の動作を示すフローチャートである。図3のステップS18においてエラーが検出された場合に、エラー処理手段#A5において本実行時エラー処理が行われる。

【0052】まず、記録メディアが不足したかどうかを調査する(ステップS64)。録画コマンドやデータの記録コマンドが送信された機能モジュールについて、メディア不足のエラーを送信したかどうかを調べることで分かる。記録メディアが不足している場合(ステップS64)、記録可能な代替モジュールの検索を行う(ステップS65)。記録可能な機能モジュールはあらかじめ指定しておくことで、代替モジュールの検索を行うことができる。検索に成功した場合(ステップS66)、検索された代替モジュールに対して予約メッセージを送信する(ステップS67)。エラー処理手段#A5が予約処理手段3に依頼メッセージを送り、予約処理手段3が代替モジュールに予約メッセージを送信する。予約が成功した場合(ステップS68)、スケジュールアクション1でソースとして指定されている機能モジュールと、代替モジュールとのストリームの接続を行う(ステップS69)。スケジュールアクション1のデータにお

17

いて、エラーになった機能モジュールを代替モジュールに置きかえることで、予約情報、接続情報、開始コマンド情報を作成することができる。接続に成功した場合（ステップS70）、代替モジュールに対して開始コマンドを送信する（ステップS71）。コマンドの送信に成功した場合（ステップS72）、ステップS16に戻り処理を続ける。

【0053】記録メディアの不足ではない場合（ステップS64）、エラー処理手段#A5ではエラー処理を行うことができないためスケジュールアクション1の停止処理を行う（ステップS76）。ストリームを切断し、予約を解放する。関係するデバイス管理モジュールにスケジュールアクション1の削除を依頼し、削除したことを見つけるイベントを発行し、スケジュールアクション1をスケジュールアクションデータベース6から削除する（ステップS77）。

【0054】上記ステップS66で代替モジュールの検索に失敗した場合、スケジュールアクション1の停止処理を行う（ステップS76）。関係するデバイス管理モジュールにスケジュールアクション1の削除を依頼し、削除したことを見つけるイベントを発行し、スケジュールアクション1をスケジュールアクションデータベース6から削除する（ステップS77）。上記ステップS68で代替モジュールの予約に失敗した場合、他に代替モジュールがあるかどうか調査する（ステップS75）。他に代替モジュールがない場合、スケジュールアクションの停止処理を行う（ステップS76）。関係するデバイス管理モジュールにスケジュールアクション1の削除を依頼し、削除したことを見つけるイベントを発行し、スケジュールアクション1をスケジュールアクションデータベース6から削除する（ステップS77）。他に代替モジュールがある場合（ステップS75）、ステップS67に戻って処理を続ける。

【0055】上記ステップS70で接続に失敗した場合、それまでに代替モジュールについて処理していた接続処理を取り消し、予約の取り消しを行う（ステップS73）。また、上記ステップS72でコマンド送信に失敗した場合、送信済みコマンドを取り消し、接続処理を取り消し、予約の取り消しを行う（ステップS74）。ステップS75以降の処理は前述の通りである。以上の実行時エラー処理を実行することにより、スケジュールアクションの実行中にエラーを検出した場合、記録メディアが不足している場合に代替モジュールを検索し、記録を続けることが可能になる。

【0056】次に、無効フラグを立てた元のスケジュールアクションが有効になった場合の処理を説明する。図7は、無効フラグを立てた元のスケジュールアクションが有効になった場合の動作を示すフローチャートである。無効フラグを立てた元のスケジュールアクションについては、被制御機器、デバイス管理モジュール、機能

18

モジュール、及びネットワークリソースの監視はスケジュールアクション処理手段4によって引き続き行われている。一旦なくなった機能モジュールがネットワーク上に再接続された場合や、使われていたデバイスが解放された場合などが考えられる。元のスケジュールアクションが有効になったことを検出すると、スケジュールアクション処理手段4はエラー処理手段#A5に通知する。エラー処理手段#A5は代替スケジュールアクションが設定されているかどうか検索する（ステップS78）。

- 10 もし、代替スケジュールアクションが設定されているれば、それを指すデータが元のスケジュールアクションに含まれている。設定されている場合、代替スケジュールアクションを無効にする（ステップS79）。関係するデバイス管理モジュール上のスケジュールアクションを削除し、代替スケジュールアクションに無効フラグを立てる。次に有効になった元のスケジュールアクションを再登録する（ステップS80）。登録方法は前述の通りである。

- 【0057】登録に成功した場合（ステップS81）、
20 元のスケジュールアクションの無効フラグを有効にする（ステップS82）。無効になった代替スケジュールアクションと対応付けを行う（ステップS83）。代替スケジュールアクションを設定したときと同じように、元のスケジュールアクションには代替スケジュールアクションのIDを指すデータを付加し、代替スケジュールアクションには元のスケジュールアクションのIDを指すデータを付加する。スケジュールアクションの監視をそのまま続ける（ステップS84）。代替スケジュールアクションの監視は中止する（ステップS85）。

- 30 【0058】上記ステップS81で元のスケジュールアクションの登録に失敗した場合、ステップS79で無効にした代替スケジュールアクションを再登録する（ステップS90）。登録に成功した場合（ステップS91）、元のスケジュールアクションとの対応付けを行い（ステップS92）、代替スケジュールアクションの監視を続ける（ステップS93）。上記ステップS91で代替スケジュールアクションの登録に失敗した場合、エラー処理手段#A5は代替処理不可と判定し（ステップS94）、元のスケジュールアクション及び代替スケジュールアクションをどちらも削除する（ステップS95）。削除したことを通知するためのイベントを発行する。

- 40 【0059】上記ステップS78で代替スケジュールアクションが設定されていない場合、有効になった元のスケジュールアクションの再登録を行う（ステップS86）。登録に成功した場合（ステップS87）、スケジュールアクションの監視を続ける（ステップS88）。登録に失敗した場合（ステップS87）、削除したことを通知するイベントを発行し、スケジュールアクションを削除する（ステップS89）。

19

【0060】以上の処理を実行することにより、代替スケジュールアクションを登録した状態でも、元のスケジュールアクションの監視を続けることで、再び実行条件を満たすようになったことを検出することができる。元のスケジュールアクションのデータから代替スケジュールアクションのIDを探し、代替スケジュールアクションを削除して元のスケジュールアクションを再登録することで、ユーザが指定した元のスケジュールアクションに戻すことができる。

【0061】次に、元のスケジュールアクションが無効になったため、代替スケジュールアクションを設定し、さらに代替スケジュールアクションが無効になった場合の処理を説明する。図8は、代替スケジュールアクションが無効になった場合の動作を示すフローチャートである。スケジュールアクション処理手段4が代替スケジュールアクションを監視した結果、無効になったことを検出した場合、無効になった代替スケジュールアクションは削除する(ステップS96)。元のスケジュールアクションのデータを検索し元のスケジュールアクションの代替スケジュールアクションの検索をし直す(ステップS97)。代替スケジュールアクションの検索については、図4のフローで説明した通りである。

【0062】代替スケジュールアクションが検索された場合(ステップS97)、代替スケジュールアクションの登録を行う。登録に成功した場合(ステップS98)、元のスケジュールアクションとの対応付けを行う(ステップS99)。スケジュールアクション処理手段4は代替スケジュールアクションの監視を行う(ステップS100)。代替スケジュールアクションの検索に失敗した場合(ステップS97)、又は登録に失敗した場合(ステップS98)、エラー処理手段#A5は代替処理不可と判定する(ステップS101)。

【0063】以上の処理を実行することにより、元のスケジュールアクションが無効になったため、代替スケジュールアクションを設定し、さらに代替スケジュールアクションが無効になったことが検出された場合、無効になった代替スケジュールアクションは削除し、元のスケジュールアクションに対する代替スケジュールアクションを再度検索することが可能になる。このように、常に元のスケジュールアクションに対する代替スケジュールアクションを検索することで、ユーザの希望に沿った代替案を作成することができる。

【0064】以上のように、第1の実施の形態のリソース管理装置は、ネットワーク上に存在する任意の装置のうち必要なモジュールを利用したアプリケーションを実行できる装置#Aと、被制御機器内の機能を管理する機能モジュールの予約処理を行う予約処理手段3と、コマンドをあらかじめ登録しておく、コマンドの実行を開始するスケジュールアクション処理手段4と、登録及び実行が行われるスケジュールアクションを保存しておくス

20

ケジュールアクションデータベース6とを備える装置#Bと、被制御機器を管理するデバイス管理モジュール#A11と、被制御機器内の機能を管理する機能モジュール#A12とを備える装置#Cと、デバイス管理モジュール#A11とを備える装置#Dとが、通信回線によって相互に接続されて構成され、装置#Bは、スケジュールアクション処理手段4によってエラーが検出された場合に、エラー処理を行うためのエラー処理手段#A5と、代替のスケジュールアクションを保存しておく代替スケジュールアクションデータベース7とを備え、装置#Cは、デバイス管理モジュール#A11に制限されたスケジュールアクションを保存しておくデバイス管理用スケジュールアクションデータベース#A16を備え、利用者が予約したスケジュールアクションを登録するステップと、登録から停止時間になるまでスケジュールアクションの実行条件を監視するステップと、検出されたエラー情報をエラー処理手段#A5に通知するステップと、エラー処理手段#A5では、検出されたエラーを受け取って代替スケジュールアクションを検索するステップと、検索した代替スケジュールアクションを代替案として登録するステップと、元のスケジュールアクションと前記代替案を関連付けして記録するステップとを順次実行するようにしたので、スケジュールアクションの登録後にエラーが検出された場合に、エラーを回避するための代替スケジュールアクションを作成することができる。また、代替スケジュールアクションを登録した状態でも、元のスケジュールアクションの監視を続けることで、再び実行条件を満たすようになったことを検出することができる。元のスケジュールアクションのデータから代替スケジュールアクションのIDを探し、代替スケジュールアクションを削除して元のスケジュールアクションを再登録することで、ユーザが指定した元のスケジュールアクションに戻すことができる。

【0065】第2の実施の形態

図9は、本発明の第2の実施の形態のネットワークシステムの全体構成を示す図である。本実施の形態の説明にあたり、図1と同一構成部分には同一符号を付して重複部分の説明を省略する。図9において、IEEE1394に準拠した通信回線によって相互に接続されたネットワーク上に、装置#A、装置#E(第2の装置)、装置#C、装置#Dがある。

【0066】図1に示す装置との差異は、装置#E内のエラー処理手段#B18及び一時記憶ディスク#A19があることである。エラー処理手段#B18は、記録に関するコマンドを処理する機能モジュールに関しては、一時記憶ディスク#A19を利用した代替モジュールの検索を行う。また、一時記憶ディスクにアクセスする機能及び予約と解放を行う機能を持つ。一時記憶ディスク#A19は、エラー処理手段#B18のみがアクセス可能なディスクであり、自分自身の予約と解放を行うAP

21

I (Application Program Interface) を持ち、エラー処理手段#B18から予約を行う。

【0067】第1の実施の形態と同様に、スケジュールアクションの実行条件を監視している間にエラーが起こった場合を考える。エラー処理手段#B18は、開始コマンドのデータから内容を解析し、記録に関するコマンドを処理する機能モジュールに関しては、代替モジュールの検索を行う時に、一時記憶ディスク#A19を利用できるかどうかの検討を行う。利用できる場合には、一時記憶ディスク#A19を利用した代替スケジュールアクションを作成し、登録を行う。登録に成功した場合には、元のスケジュールアクションとの対応付けを行い、代替スケジュールアクションを監視する。エラー検出時にスケジュールアクションが開始していた場合には、代替スケジュールアクションを開始する。エラー検出時にスケジュールアクションが実行されていた場合には、代替スケジュールアクションを実行する。

【0068】一時記憶ディスク#A19が利用できない場合は、ネットワーク上に他の代替となる機能モジュールがあるかどうか検索を行い、第1の実施の形態と同様に処理を行う。一時記憶ディスク#A19は、エラー処理手段#B18のみアクセス可能であるから、他のスケジュールアクションと重なる可能性が低い。そのため、代替モジュールとして利用する際の確実性が高まり、エラー処理手段#B18の処理能力が向上する。代替モジュールとして一時記憶ディスク#A19を利用する場合、代替スケジュールアクションには一時記憶ディスクを表す識別子を用いてデータを記録する。一時記憶ディスク#A19は、自分自身の予約と解放を行うAPIを持ち、エラー処理手段#B18から予約を行うことができるようになる。代替スケジュールアクションを実行する際には、機能モジュールの予約を行なう際に、一時記憶ディスク#A19の予約を行い、利用する。その他の代替スケジュールアクションの登録・削除の処理については第1の実施の形態と同様である。

【0069】また、代替スケジュールアクションを実行して完了した後、元のスケジュールアクションで指定された機能モジュールがネットワーク上に接続された場合、エラー処理手段#B18は、一時記憶ディスク#A19に記憶したデータを指定された機能モジュールに転送しておくことができる。このようにすることで、ユーザがあらかじめ指定したスケジュールアクションの通りに実行結果を作成することが可能である。

【0070】以上のように、第2の実施の形態のリソース管理装置は、装置#Eが、代替スケジュールアクションで指定された内容を記憶する一時記憶ディスク#A19と、機能モジュールの代替となる代替モジュールの検索と代替スケジュールアクションの作成及び登録と一時記憶ディスク#A19にアクセスする機能及び予約と解放を行なうエラー処理手段#B18とを備えているので、

22

一時記憶ディスク#A19はエラー処理手段#B18からのみアクセス可能であるため、エラー処理の時のみ利用される。したがって、他のスケジュールアクションと重なる可能性が低いため、代替モジュールとして利用する際の確実性が高まり、エラー処理手段#B18の処理能力が向上する。また、一時記憶ディスク#A19に記録したデータを元のスケジュールアクションで指定されたモジュールへ転送する機能を持たせることにより、ユーザが指定した元のスケジュールアクションを実行することが可能となる。

【0071】第3の実施の形態

図10は、本発明の第3の実施の形態のネットワークシステムの全体構成を示す図である。本実施の形態の説明にあたり、図1と同一構成部分には同一符号を付して重複部分の説明を省略する。図10において、IEEE1394に準拠した通信回線によって相互に接続されたネットワーク上に、装置#A、装置#B、装置#C、装置#D、装置#F、装置#G（第3の装置）がある。

【0072】図1に示す装置との差異は、装置#B内のエラー処理手段#C20を持ち、装置#Fには、一時記憶ディスク#B21を持つことである。装置#Gは、一時記憶ディスク#B21のデバイスを制御するデバイス管理モジュール#B22、一時記憶ディスク#B21の機能を制御する機能モジュール#B23、デバイス管理用スケジュールアクションデータベース#B24、装置#G内のデータについてデータ変換を行いデータの送受信を管理する通信手段#G25を備える。装置#G内のモジュールは他の装置上に存在していても構わない。

【0073】エラー処理手段#C20は、記録に関するコマンドを処理する機能モジュールに関しては、代替モジュールの検索を行う時に、ネットワーク上に一時記憶ディスクが存在するかどうか検索する機能を持つ。一時記憶ディスク#B21がネットワーク上に存在する場合、これを用いた代替スケジュールアクションの検索を行う。

一時記憶ディスク#B21のスケジュールアクションの状況は、デバイス管理モジュール#B22に問い合わせ、デバイス管理用スケジュールアクションデータベース#B24を参照することで検索することができる。利用できる場合には、機能モジュール#B23を用いて代替スケジュールアクションを作成し、登録を行う。登録に成功した場合は、代替スケジュールアクションと元のスケジュールアクションを対応付けし、それぞれ代替スケジュールアクションデータベース7とスケジュールアクションデータベース6に記録しておく。一時記憶ディスク#B21が利用できない場合や代替スケジュールアクションの登録に失敗した場合は、第1の実施の形態で説明した処理と同様に代替スケジュールアクションを検索する処理を行う。

【0074】以上のように、第3の実施の形態のリソース管理装置は、さらに、一時記憶ディスク#B21のデ

23

バイスを制御するデバイス管理モジュール#B22と、一時記憶ディスク#B21の機能を制御する機能モジュール#B23、デバイス管理用スケジュールアクションデータベース#B24と、装置#Bの外部のネットワーク上に接続された装置#Fである一時記憶ディスク#B21とを備えて構成したので、利用者が予約した機器に不具合があった場合に、代替案を示すことができる。

【0075】特に、一時記憶ディスク#B21としてランダムアクセス可能なディスクを採用すれば、登録の順番を気にすることなく代替スケジュールアクションを作成することが可能である。一時記憶ディスク#B21として、複数の装置から同時にアクセス可能なディスクを採用すれば、いくつかのスケジュールアクションが重なっていたとしても、同時に処理することが可能となりスケジュールアクション及び代替スケジュールアクションの実行可能性が大幅に向かう。

【0076】第4の実施の形態

図11は、本発明の第4の実施の形態のネットワークシステムの全体構成を示す図である。本実施の形態の説明にあたり、図1と同一構成部分には同一符号を付して重複部分の説明を省略する。図11において、IEEE1394に準拠した通信回線によって相互に接続されたネットワーク上に、装置#A、装置#B、装置#C、装置#D、装置#H（第5の装置）がある。

【0077】図1に示す装置との差異は、装置#B内に他の装置からのエラー処理依頼を受けてスケジュールアクションのエラー処理を行うエラー処理手段#D26を持ち、装置#Hが以下の構成を持つことである。装置#Hは、利用するモジュールのために機能モジュールの予約処理を行う予約処理手段3、開始時間・停止時間・必要なモジュール・送信するコマンドをあらかじめ登録しておき、実行可能な状態を監視しておき、開始時間になると実行を開始するスケジュールアクション処理手段4、スケジュールアクション処理手段4によって登録や実行が行われるスケジュールアクションを記録しておくスケジュールアクションデータベース6、スケジュールアクションの実行ができなくなるようなエラーをスケジュールアクション処理手段4が検出した場合、他の装置の持つエラー処理手段#D26に対してエラー処理を依頼するエラー処理依頼手段27、装置#H内のデータについてデータ変換を行いデータの送受信を管理する通信手段#H28を備える。

【0078】装置#H内のスケジュールアクション処理手段4では、アプリケーション1からのメッセージによってスケジュールアクションの登録を行い、データをスケジュールアクションデータベース6に保存する。登録方法は第1の実施の形態と同様である。スケジュールアクション処理手段4は、登録されたスケジュールアクションの実行条件を監視している。

【0079】データの開始時間までの間又は実行中にエ

24

ラーを検出した場合、スケジュールアクション処理手段4は自身の装置内のエラー処理依頼手段27にスケジュールアクションの情報とエラー内容を知らせる。また、元のスケジュールアクションに関するデバイス管理モジュールへの登録を削除し、無効フラグを立てる。

【0080】エラー処理依頼手段27は、ネットワーク上でエラー処理手段#D26の検索を行い、装置#B内のモジュールを探し出す。情報管理モジュール17内に情報を登録することで検索を行う。又は、エラー処理依頼手段27からエラー処理手段#Dを検索するメッセージを送信し、エラー処理手段#Dから返事を受け取ることも可能である。エラー処理依頼手段27は、エラー処理手段#D26に対し、スケジュールアクションの代替スケジュールアクションを作成するよう依頼するメッセージを送信する。この時、スケジュールアクションのデータ及びエラー内容を通知する。

【0081】エラー処理手段#D26は、依頼を受けると、エラー内容に応じて代替スケジュールアクションを検索する。検索方法は第1の実施の形態と同様の方法で行える。代替スケジュールアクションが見つかったら、代替スケジュールアクションを登録し、装置#B内の代替スケジュールアクションデータベース7に保存する。

代替スケジュールアクションには、装置#HのID及び装置#H内のスケジュールアクションデータベース6にある元のスケジュールアクションのIDを記録しておき、対応付けを行う。エラー処理手段#D26は、代替スケジュールの検索と登録に成功したかどうかの返事を返信し、代替スケジュールアクションのIDを記録しておく。装置#H内の元のスケジュールアクションには、装置#BのID及び代替スケジュールアクションのIDと代替案があることを示すフラグを記録しておく。

【0082】元のスケジュールアクションについては装置#H内のスケジュールアクション処理手段4が監視を続けている。再び元のスケジュールアクションが実行可能になった場合、エラー処理依頼手段27に実行可能になったことを通知する。代替スケジュールアクションが設定されいれば、装置#B内のエラー処理手段#D26に対して実行可能になったことを通知し、代替スケジュールアクションを無効にするよう依頼する。別の装置

40 内に代替スケジュールアクションを保存してある場合は、元のスケジュールアクションの登録に成功すると、代替スケジュールアクションのデータは削除するよう依頼する。元のスケジュールアクションの登録に失敗すると、代替スケジュールアクションを再登録し、スケジュールアクション処理手段4で元のスケジュールアクションについては監視を続ける。

【0083】代替スケジュールアクションが無効になった場合、装置#B内のスケジュールアクション処理手段4がエラーを検出する。代替スケジュールアクションデータベース6からデータを削除し、代替スケジュールア

50

25

クションの元のスケジュールアクションのエラー処理依頼手段27に対して、データを削除したことを通知する。エラー処理依頼手段27は、元のスケジュールアクションから代替スケジュールアクションの情報を削除し、再度代替スケジュールアクションの作成をエラー処理手段#D26に依頼する。また、一度代替スケジュールアクションの作成に失敗すると、元のスケジュールアクションを無効にするように設定してもよい。

【0084】以上のように、第4の実施の形態のリソース管理装置は、さらに、機能モジュールの予約処理を行う予約処理手段3、スケジュールアクションを記録しておくスケジュールアクションデータベース6と、ネットワークに接続されている他の装置が持つエラー処理手段に対してエラー処理を依頼するエラー処理依頼手段27とから構成される装置#Hを備えて構成したので、ネットワーク上にあるすべてのスケジュールアクション処理手段4がエラー処理手段#D26及び代替スケジュールアクションデータベースを持つ必要がなく、プログラムの効率を高め、ネットワーク上で機能を分散することができる。

【0085】第5の実施の形態

図12は、本発明の第5の実施の形態のネットワークシステムの全体構成を示す図である。本実施の形態の説明にあたり、図1及び図11と同一構成部分には同一符号を付して重複部分の説明を省略する。図12において、IEEE1394に準拠した通信回線によって相互に接続されたネットワーク上に、装置#A、装置#E（第2の装置）、装置#C、装置#D、装置#Hがある。

【0086】図1に示す装置との差異は、装置#E内にエラー処理手段#E29及び一時記憶ディスク#C30を持つことである。エラー処理手段#E29は、記録に関するコマンドを処理する機能モジュールに関しては、代替モジュールの検索を行う時に一時記憶ディスク#C30を利用した代替スケジュールアクションを検索する。また、一時記憶ディスク#C30にアクセスする機能と、予約と解放を行う機能を持つ。一時記憶ディスク#C30は、エラー処理手段#E29のみがアクセス可能なディスクであり、自分自身の予約と解放を行うAPIを持ち、エラー処理手段#E29から予約を行うことができる。

【0087】第4の実施の形態と同様に装置#H内のスケジュールアクション処理手段4が登録されたスケジュールアクションのエラーを検出した場合を考える。スケジュールアクション処理手段4は自身の装置内のエラー処理依頼手段27にスケジュールアクションの情報とエラー内容を知らせる。また、元のスケジュールアクションに関するデバイス管理モジュールへの登録を削除し、無効フラグを立てる。

【0088】エラー処理依頼手段27は、ネットワーク上のエラー処理手段#E29を探し、スケジュールアク

10

26

ションの代替スケジュールアクションを作成するよう依頼するメッセージを送信する。エラー処理手段#E29は、依頼を受けると、開始コマンドのデータから内容を解析し、記憶に関するコマンドを処理する機能モジュールに関しては、代替モジュールの検索を行う時に、一時記憶ディスク#C30を利用できるかどうかの検討を行う。利用できる場合には、一時記憶ディスク#C30を利用した代替スケジュールアクションを作成し、登録を行う。登録に成功した場合には、装置#HのID及び元のスケジュールアクションのIDとの対応付けを行い、代替スケジュールアクションを監視する。エラー検出時にスケジュールアクションが開始されていた場合には、代替スケジュールアクションを開始する。エラー検出時にスケジュールアクションが実行していた場合には、代替スケジュールアクションを実行する。一時記憶ディスク#C30が利用できない場合は、ネットワーク上に他の代替となる機能モジュールがあるかどうか検索を行い、請求項4と同様に処理を行う。

【0089】代替モジュールとして一時記憶ディスク#C30を利用する場合、代替スケジュールアクションには一時記憶ディスクを表す識別子を用いてデータを記録する。一時記憶ディスク#C30は、自分自身の予約と解放を行うAPIを持ち、エラー処理手段#E29から予約を行う。代替スケジュールアクションを実行する際には、機能モジュールの予約を行う際に、一時記憶ディスク#C30の予約を行い、利用する。その他の代替スケジュールアクションの登録・削除の処理については第4の実施の形態と同様である。

【0090】また、代替スケジュールアクションを実行して完了した後、元のスケジュールアクションで指定された機能モジュールがネットワーク上に接続された場合、エラー処理手段#E29は、一時記憶ディスク#C30に記憶したデータを指定された機能モジュールに転送しておくことができる。このようにすることで、ユーザーがあらかじめ指定したスケジュールアクションの通りに実行結果を作成することが可能である。装置#B上の一時記憶ディスク#C30は、装置#Bとは別にネットワークに接続するような構成にしてもよい。この場合、第3の実施の形態と同様に、エラー処理手段#Eに、一時記憶ディスクを検索する機能を持たせる。

【0091】以上のように、第5の実施の形態のリソース管理装置は、装置#Eが、代替モジュールの検索を行う時に一時記憶ディスク#C30を利用した代替スケジュールアクションを検索するエラー処理手段#E29と、一時記憶ディスク#C30とを備えて構成したので、第4の実施の形態と同様の効果を得ることができるとともに、エラー処理手段#E29を持つ端末内に一時記憶ディスク#C30が存在することで、エラー処理手段#E29の処理能力を向上させることができる。

【0092】なお、上記各実施の形態に係るネットワー

50

27

クシステムにおけるリソース管理装置は、複数の装置を通信回線で接続した例であるが、複数の機器が通信回線で相互に接続されて構成されるネットワークシステムであればどのようなネットワークであってもよい。例えば、ネットワークシステムとして、HAVi仕様書に基づく家庭内AVネットワークに適用することができる。

【0093】また、記録装置の種類や記録方法等は限定されず、全ての装置に適用可能である。例えば、記録装置として、VTR(Video Tape Recorder)のほか、HDDやDVDに記録するものでもよい。特に、データ転送速度やランダム・アクセス性能が高く、高速アクセスが可能なHDDを記録媒体に用いれば、長時間記録した番組の中から所望の番組を瞬時に検索し、直ぐに利用することが可能になる。また、光磁気ディスク等HDD以外の記録装置を用いてもよく、同様の効果を得ることができる。さらに、上記リソース管理装置を構成する各手段、例えば通信手段、エラー処理手段等の種類、数及び接続方法などは前述した実施の形態に限られない。

【0094】以上説明したリソース管理装置及びその方法は、このリソース管理装置及びその方法を機能させるためのプログラムでも実現される。このプログラムはコンピュータで読み取り可能な記録媒体に格納されている。本発明では、この記録媒体として、メインメモリそのものがプログラムメディアであってもよいし、また外部記憶装置としてプログラム読み取り装置が設けられ、そこに記録媒体を挿入することで読み取り可能なプログラムメディアであってもよい。いずれの場合においても、格納されているプログラムはCPUがアクセスして実行させる構成であってもよいし、あるいはいずれの場合もプログラムを読み出し、読み出されたプログラムは、図示されていないプログラム記憶エリアにダウンロードされて、そのプログラムが実行される方式であってもよい。このダウンロード用のプログラムは予め本体装置に格納されているものとする。

【0095】ここで、上記プログラムメディアは、本体と分離可能に構成される記録媒体であり、磁気テープやカセットテープ等のテープ系、フロッピー(登録商標)ディスクやハードディスク等の磁気ディスクやCD-R OM/MO/MD/DVD等の光ディスクのディスク系、ICカード/光カード等のカード系、あるいはマスクROM、EPROM、EEPROM、フラッシュROM等による半導体メモリを含めた固定的にプログラムを担持する媒体であってもよい。

【0096】さらに、図示されていないが、外部の通信ネットワークとの接続が可能な手段を備えている場合には、その通信接続手段を介して通信ネットワークからプログラムをダウンロードするように、流動的にプログラムを担持する媒体であってもよい。なお、このように通信ネットワークからプログラムをダウンロードする場合には、そのダウンロード用プログラムは予め本体装置に

28

格納しておくか、あるいは別な記録媒体からインストールされるものであってもよい。なお、記録媒体に格納されている内容としてはプログラムに限定されず、データであってもよい。

【0097】

【発明の効果】以上、詳述したように、本発明によれば、利用者が予約した機器に不具合があった場合に、代替案を示すことができるリソース管理装置及びリソース管理方法を提供することができる。請求項1及び2の発明では、スケジュールアクションの登録後にエラーが検出された場合に、エラーを回避するための代替スケジュールアクションを作成することが可能である。また、代替スケジュールアクションを登録した状態でも、元のスケジュールアクションの監視を続けることで、再び実行条件を満たすようになったことを検出することができる。

【0098】請求項3の発明では、一時記憶ディスクはエラー処理手段からのみアクセス可能であるため、エラー処理の時のみ利用される。したがって、他のスケジュールアクションと重なる可能性が低いため、代替モジュールとして利用する際の確実性が高まり、エラー処理手段の処理能力が向上する。また、一時記憶ディスクに記録したデータを元のスケジュールアクションで指定されたモジュールへ転送する機能を持たせることにより、ユーザーが指定した元のスケジュールアクションを実行することができる。

【0099】請求項4の発明では、ネットワーク上に一時記憶ディスクを備えた装置を接続するため、エラー処理手段を持つ端末内に十分な記憶容量が確保できなくても代替スケジュールアクションが処理可能であり、端末のデータ記憶領域が圧迫されない。一時記憶ディスクとしてランダム・アクセス可能なディスクを採用することで、登録の順番を気にすることなく代替スケジュールアクションを作成することが可能である。また、複数の装置から同時にアクセス可能なディスクを採用すれば、いくつかのスケジュールアクションを同時に処理することが可能となり、スケジュールアクション及び代替スケジュールアクションの実行可能性が向上する。

【0100】請求項5の発明では、ネットワーク上にあるすべてのスケジュール処理手段がエラー処理手段及び代替スケジュールアクションデータベース、一時記憶ディスクを持つ必要がなく、プログラムの効率が良い。また、エラー処理手段を持つ端末内に一時記憶ディスクが存在することで、エラー処理手段の処理能力が向上する。

【0101】請求項6の発明では、ネットワーク上にあるすべてのスケジュールアクション処理手段がエラー処理手段及び代替スケジュールアクションデータベースを持つ必要がなく、プログラムの効率が良い。ネットワーク上で機能を分散することができる。

【0102】請求項7乃至10至の発明では、スケジュールアクションの監視時にエラーを検出した場合に、エラーを回避するための代替スケジュールアクションを作成することが可能となる。また、代替スケジュールアクションを登録した状態でも、元のスケジュールアクションの監視を続けることで、再び実行条件を満たすようになったことを検出することができる。元のスケジュールアクションのデータから代替スケジュールアクションのIDを探し、代替スケジュールアクションを削除して元のスケジュールアクションを再登録することで、ユーザが指定した元のスケジュールアクションに戻すことができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態のネットワークシステムの全体構成を示す図である。

【図2】本実施の形態のリソース管理装置のスケジュールアクションの登録の実行の動作を示すフローチャートである。

【図3】本実施の形態のリソース管理装置のスケジュールアクションの登録の実行の動作を示すフローチャートである。

【図4】本実施の形態のリソース管理装置の監視時エラー処理の動作を示すフローチャートである。

【図5】本実施の形態のリソース管理装置の開始時エラー処理の動作を示すフローチャートである。

【図6】本実施の形態のリソース管理装置の実行時エラー処理の動作を示すフローチャートである。

【図7】本実施の形態のリソース管理装置の無効フラグを立てた元のスケジュールアクションが有効になった場合の動作を示すフローチャートである。

【図8】本実施の形態のリソース管理装置の代替スケジュールアクションが無効になった場合の動作を示すフローチャートである。

【図9】本発明の第2の実施の形態のネットワークシステムの全体構成を示す図である。

【図10】本発明の第3の実施の形態のネットワークシステムの全体構成を示す図である。

【図11】本発明の第4の実施の形態のネットワークシステムの全体構成を示す図である。

【図12】本発明の第5の実施の形態のネットワークシステムの全体構成を示す図である。

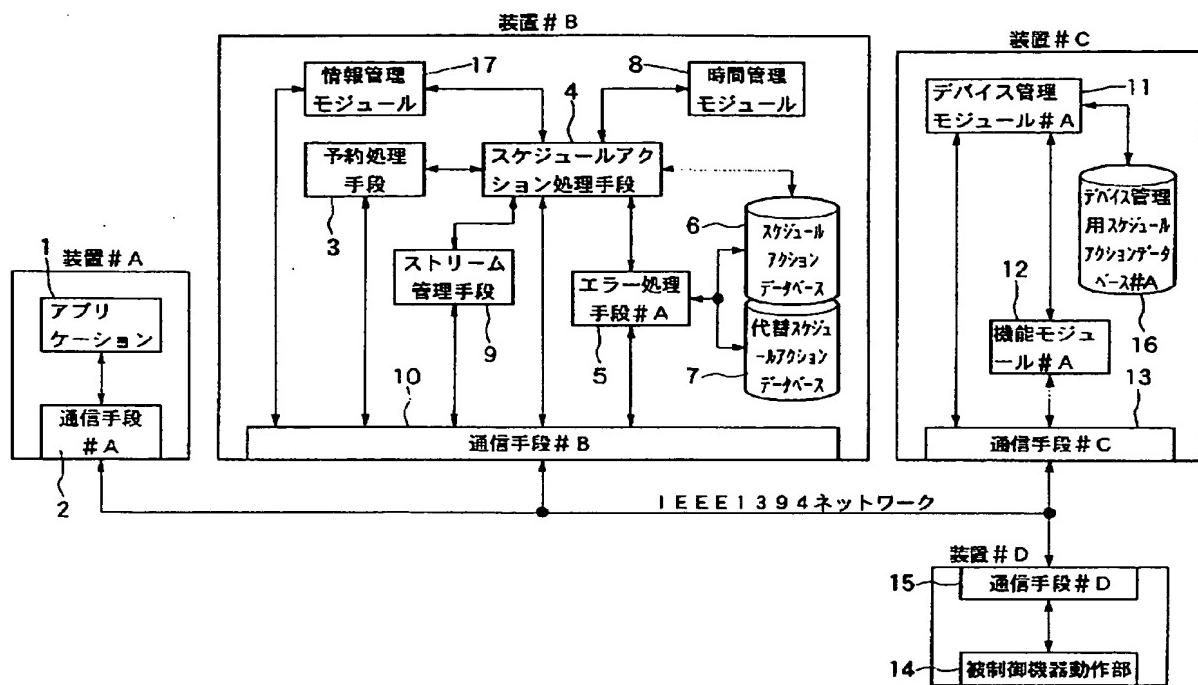
【図13】従来のネットワークを構成しない制御機器の構成を示すブロック図である。

【図14】従来のネットワークによって構成される制御機器の構成を示すブロック図である。

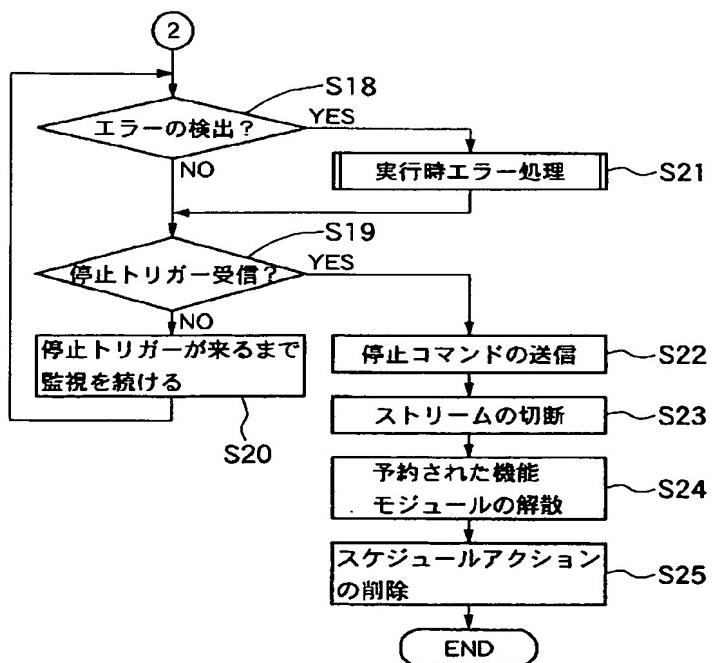
【符号の説明】

- | | |
|----|----------------------------|
| 1 | アプリケーション |
| 2 | 通信手段#A |
| 10 | 予約処理手段 |
| 4 | スケジュールアクション処理手段 |
| 5 | エラー処理手段#A |
| 6 | スケジュールアクションデータベース |
| 7 | 代替スケジュールアクションデータベース |
| 8 | 時間管理モジュール |
| 9 | ストリーム管理手段 |
| 10 | 通信手段#B |
| 11 | デバイス管理モジュール#A |
| 12 | 機能モジュール#A |
| 20 | 通信手段#C |
| 14 | 被制御機器動作部 |
| 15 | 通信手段#D |
| 16 | デバイス管理用スケジュールアクションデータベース#A |
| 17 | 情報管理モジュール |
| 18 | エラー処理手段#B |
| 19 | 一時記憶ディスク#A |
| 20 | エラー処理手段#C |
| 21 | 一時記憶ディスク#B |
| 30 | デバイス管理モジュール#B |
| 22 | 機能モジュール#B |
| 24 | デバイス管理用スケジュールアクションデータベース#B |
| 25 | 通信手段#G |
| 26 | エラー処理手段#D |
| 27 | エラー処理依頼手段 |
| 28 | 通信手段#H |
| 29 | エラー処理手段#E |
| 30 | 一時記憶ディスク#C |

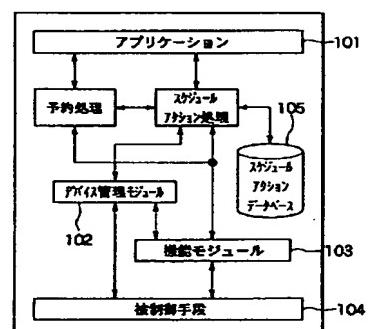
【図1】



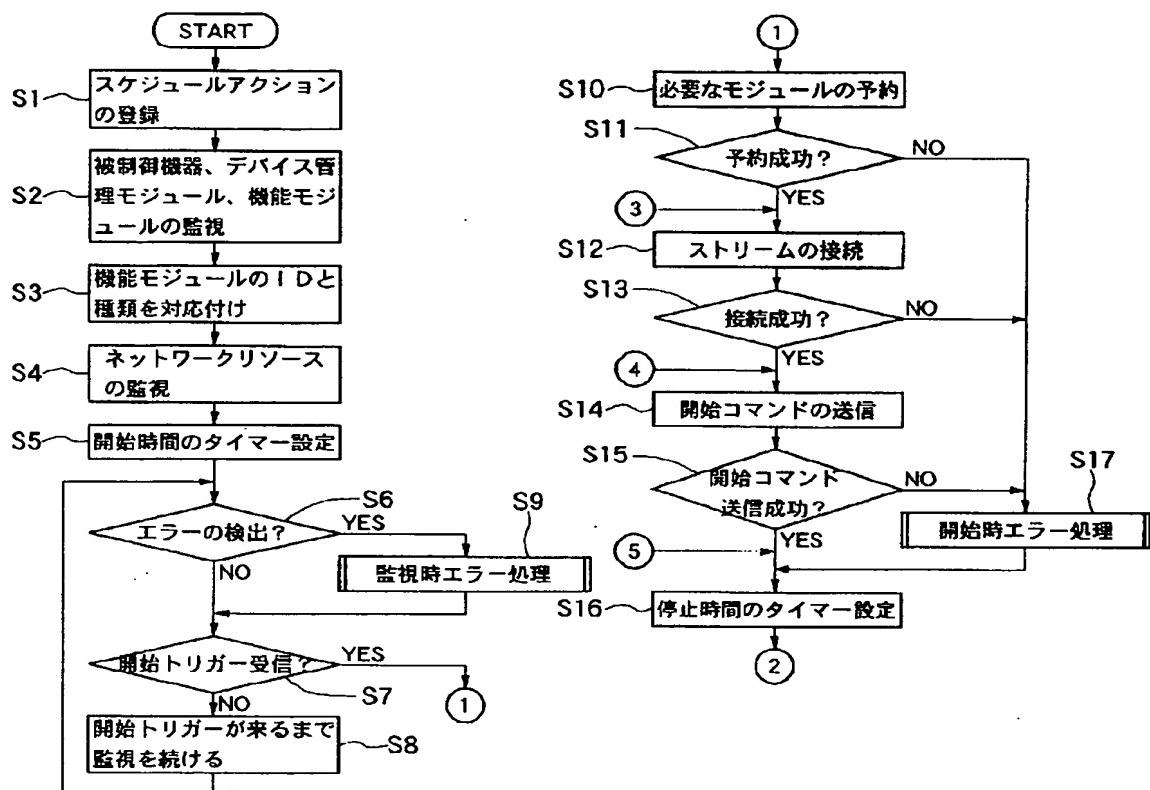
【図3】



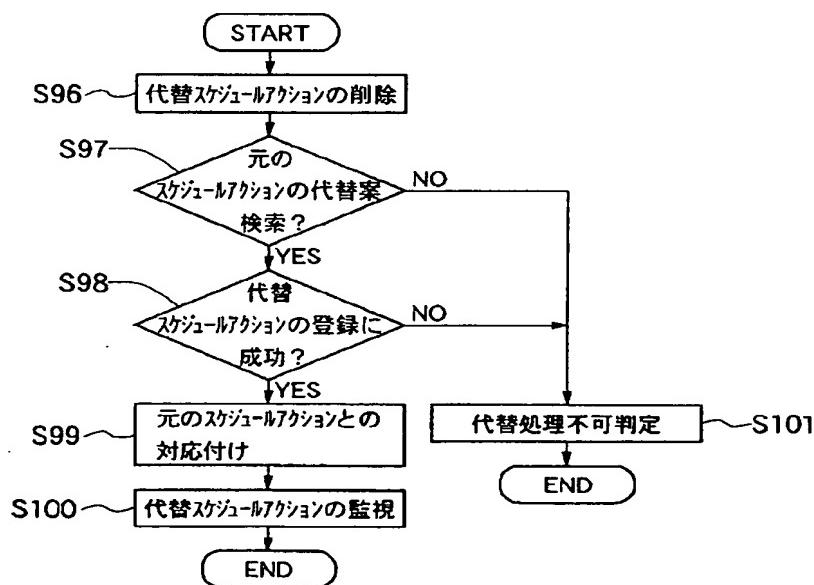
【図13】



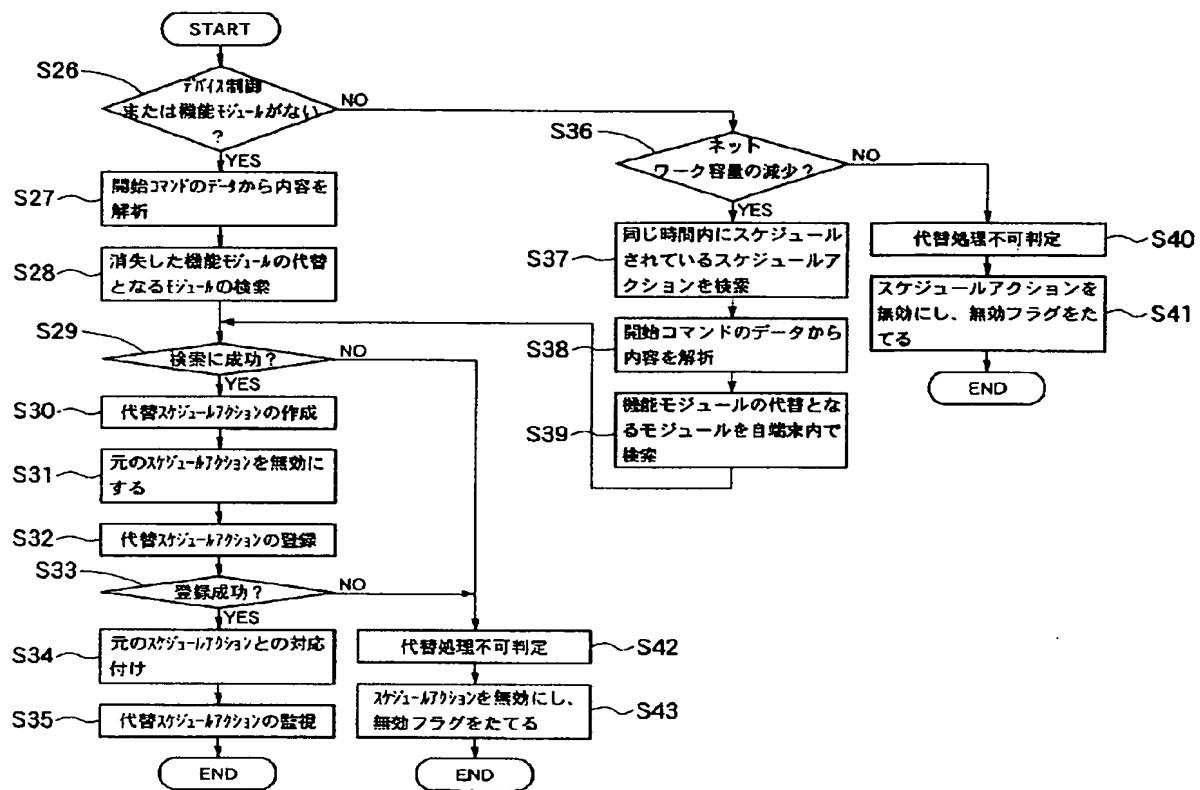
【図2】



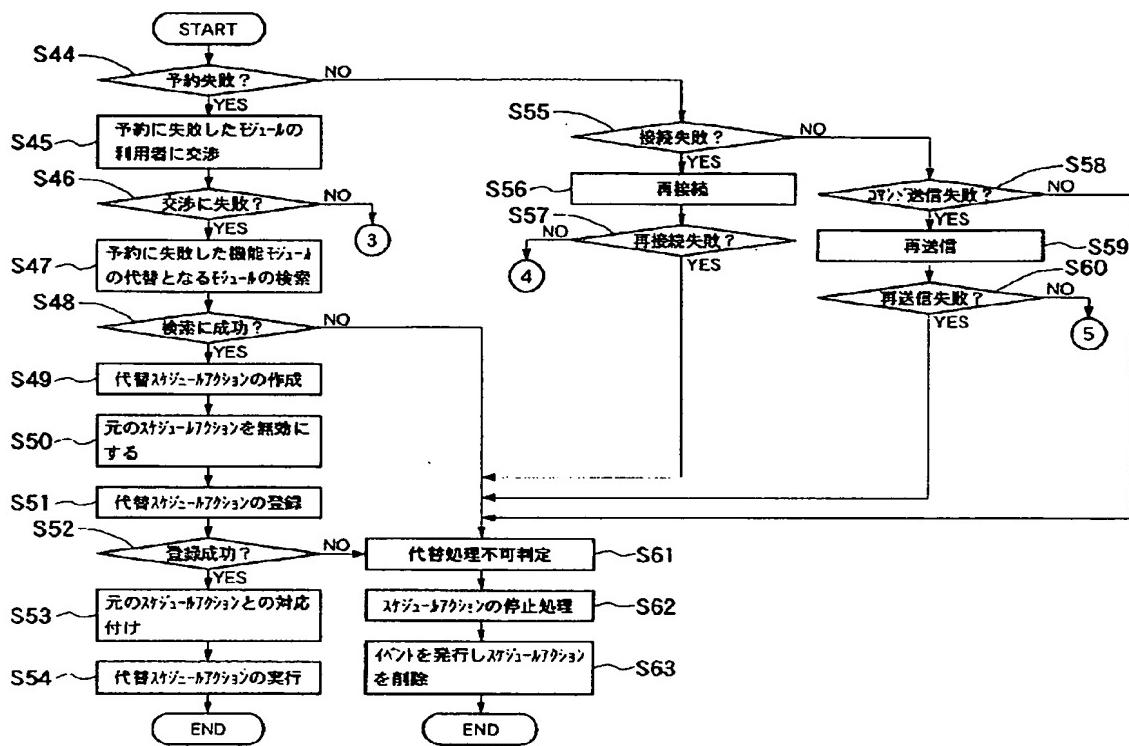
【図8】



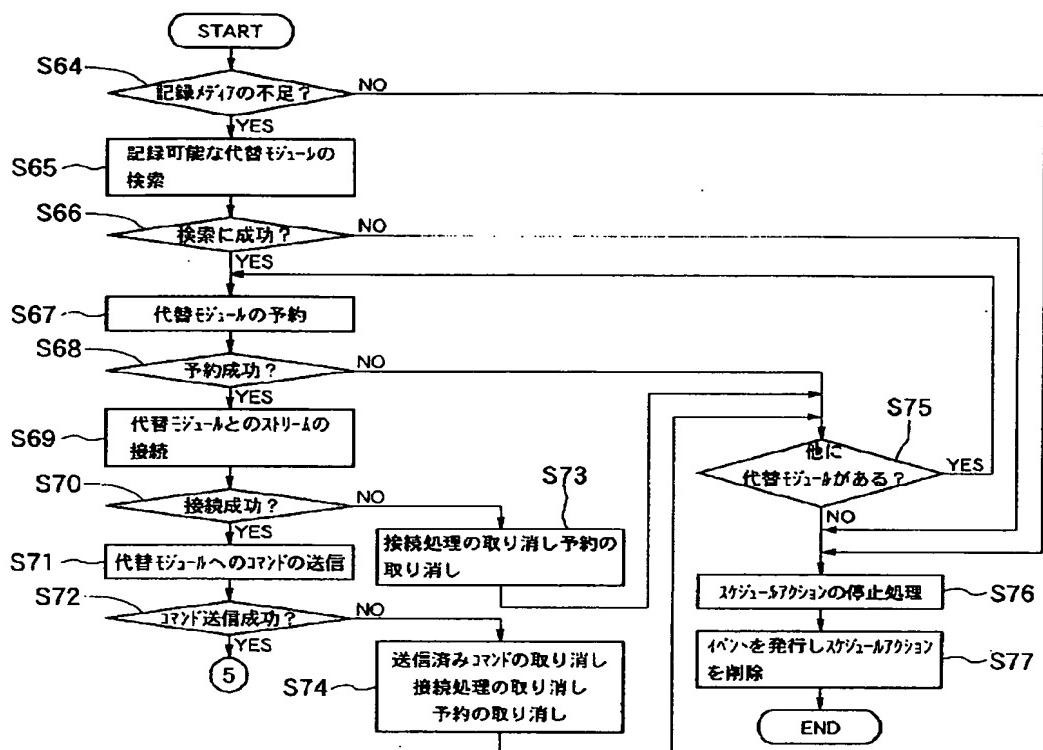
【図4】



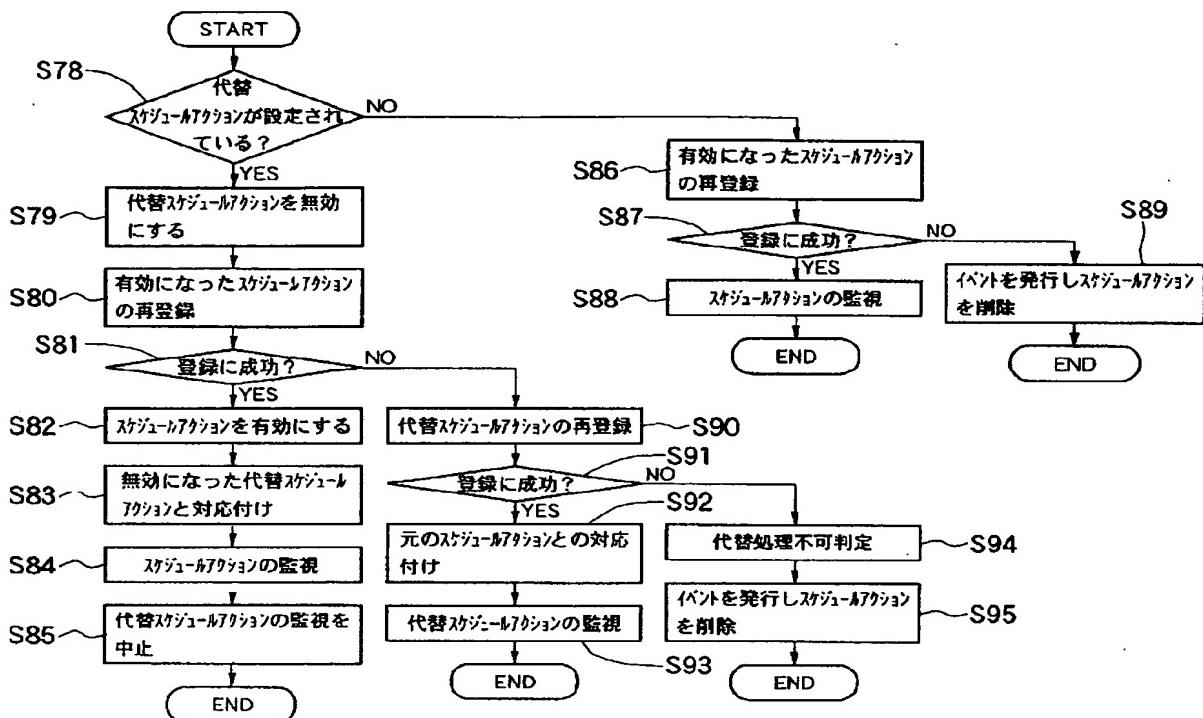
【図5】



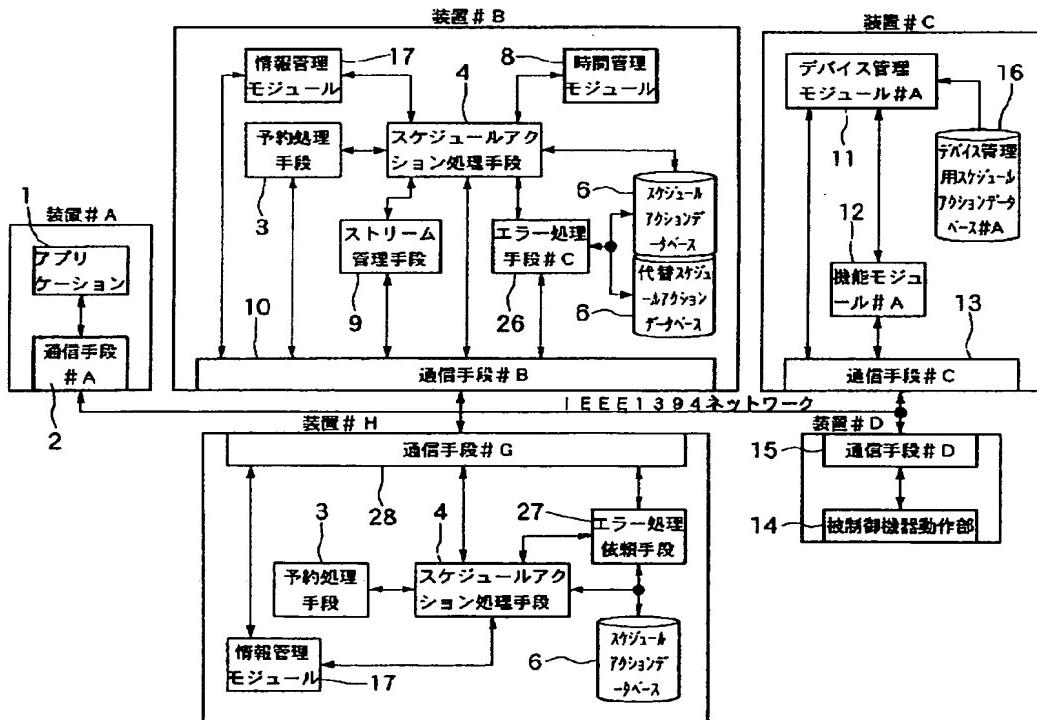
【図6】



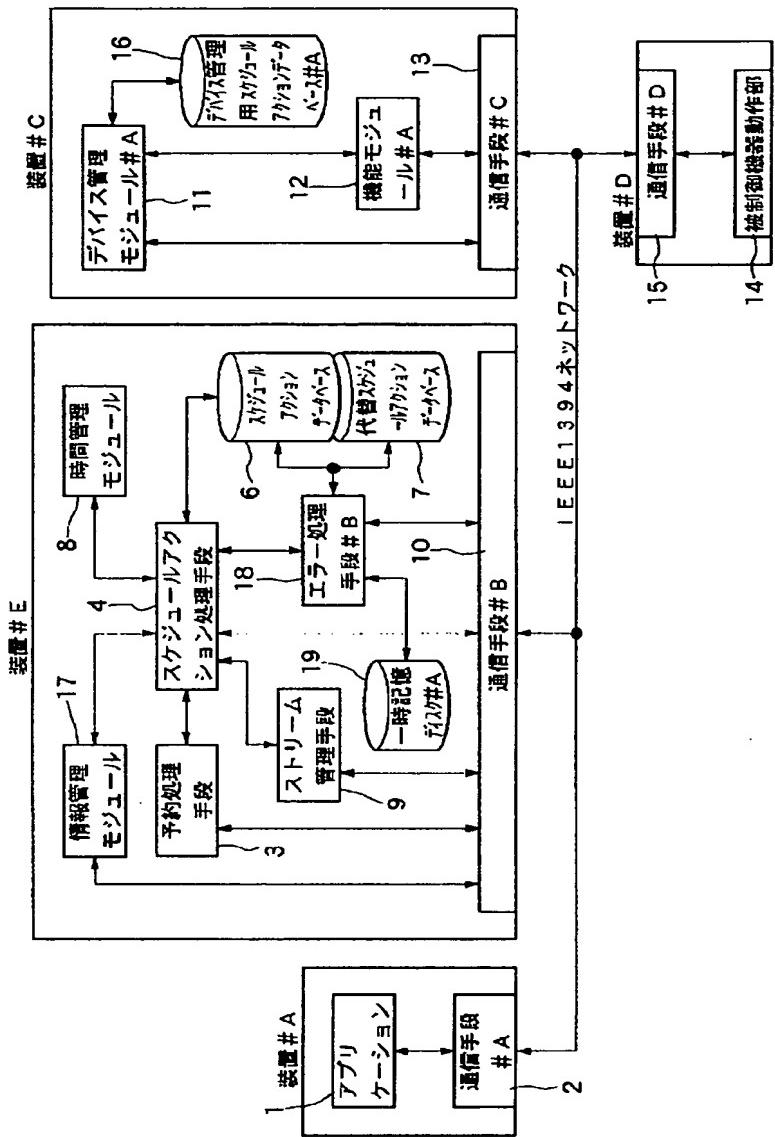
〔図7〕



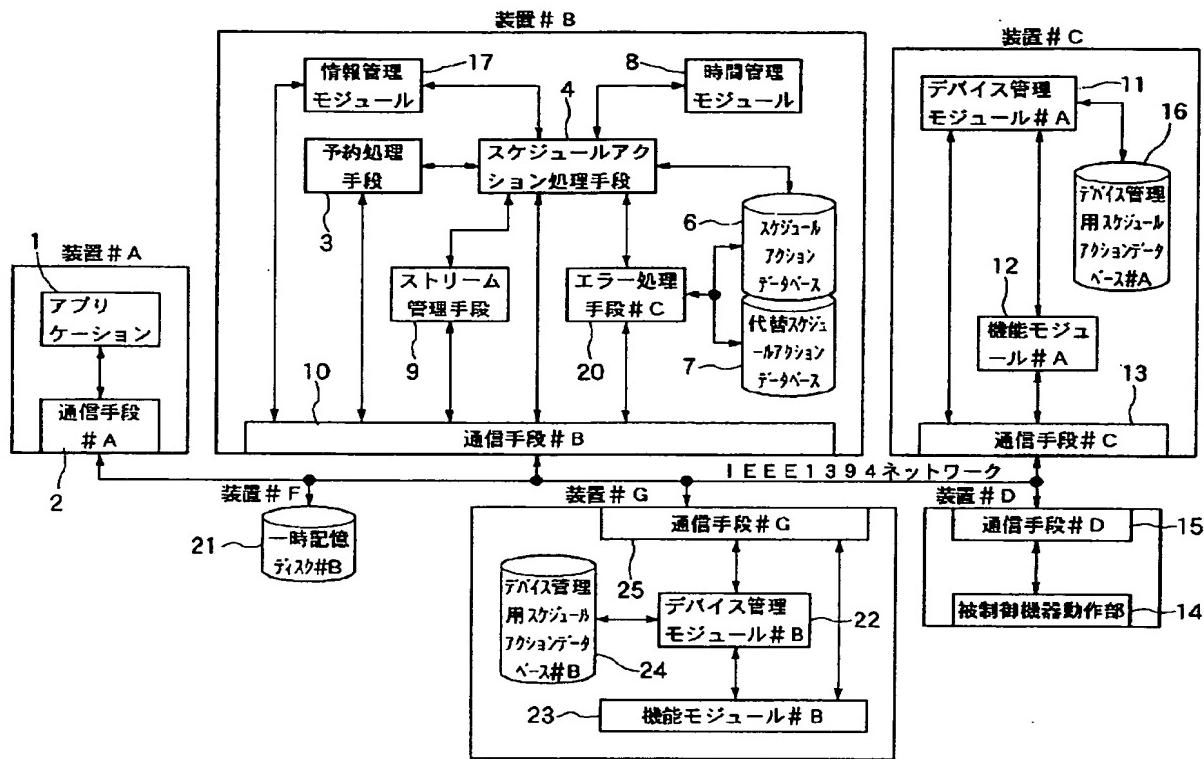
【図11】



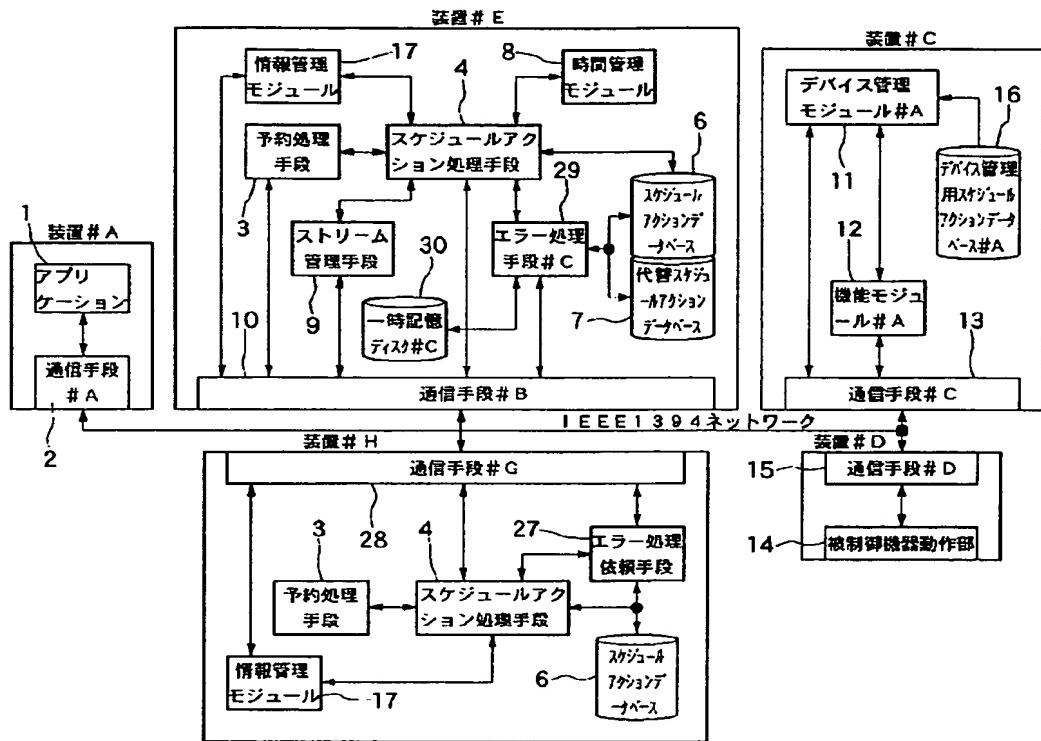
〔図9〕



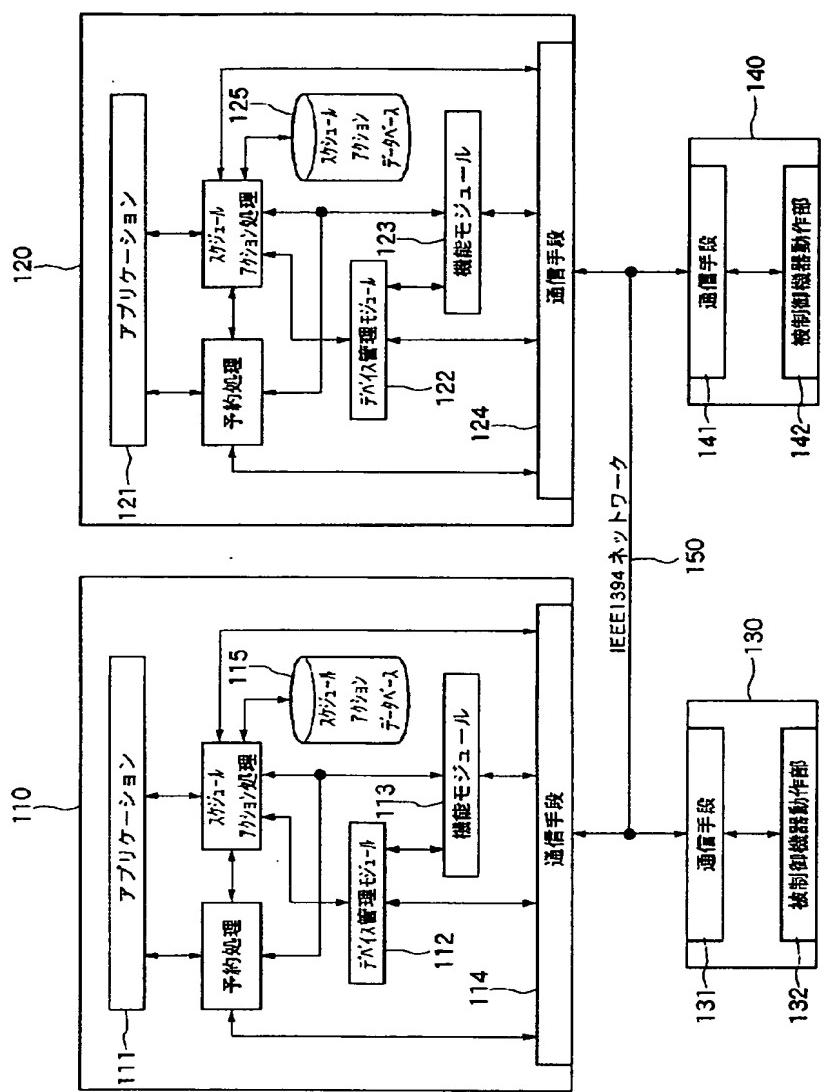
【図10】



【図12】



【図14】



フロントページの続き

(51)Int.Cl.

識別記号

F I
H 0 4 N 5/91

テーマコード(参考)

L